



AFRL-RI-RS-TR-2016-279

SOLUTIONS FOR CODING SOCIETAL EVENTS

RAYTHEON BBN TECHNOLOGIES CORP.

DECEMBER 2016

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Intelligence Advanced Research Projects Agency (IARPA) Public Release Center and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2016-279 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

ALEKSEY PANASYUK
Work Unit Manager

/ S /

MICHAEL J. WESSING
Deputy Chief, Information Intelligence
Systems & Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) DECEMBER 2016		2. REPORT TYPE FINAL TECHNICAL REPORT		3. DATES COVERED (From - To) SEP 2015 – SEP 2016	
4. TITLE AND SUBTITLE SOLUTIONS FOR CODING SOCIETAL EVENTS				5a. CONTRACT NUMBER FA8750-15-C-0276	
				5b. GRANT NUMBER N/A	
				5c. PROGRAM ELEMENT NUMBER N/A	
6. AUTHOR(S) Elizabeth Boschee				5d. PROJECT NUMBER SCSE	
				5e. TASK NUMBER IA	
				5f. WORK UNIT NUMBER RP	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Raytheon BBN Technologies Corp. 10 Moulton Street Cambridge, MA 02138				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Air Force Research Laboratory/RIEA 525 Brooks Road Rome NY 13441-4505 </div> <div style="width: 45%;"> IARPA Gate 5, 1000 Colonial Farm Rd McLean, VA 22101 </div> </div>				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI	
				11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2016-279	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. IARPA PA # 2016-00139 Date Cleared: 3 Nov 2016					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The primary goal of the IARPA Solutions for Coding Societal Events (SCSE) seedling effort was to help with known and unknown event discovery. Discovering events from news and social media is becoming a critical source of intelligence. This effort attempted to (1) develop a prototype system for novel event discovery, (2) develop a prototype system for civil unrest event extraction, and (3) engineer BBN ACCENT (ACCurate Events from Natural Text) to support broad use by the research community.					
15. SUBJECT TERMS Event discovery, civil unrest event detection, event ontology generation, unsupervised relation-learning, phrase clustering.					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT U	18. NUMBER OF PAGES 75	19a. NAME OF RESPONSIBLE PERSON ALEKSEY PANASYUK
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) N/A

Table of Contents

List of Figures	iii
List of Tables.....	iv
1 Summary	1
2 Introduction.....	2
2.1 Novel Event Class Discovery	2
2.2 Civil Unrest	4
2.3 BBN ACCENT	8
3 Methods, Assumptions, and Procedures.....	8
3.1 Novel Event Class Discovery	8
3.1.1 Event Similarity	8
3.1.2 Event/Non-Event Classification.....	13
3.1.3 Clustering.....	15
3.1.4 Experiment Data.....	16
3.2 Civil Unrest	19
3.2.1 Overview	19
3.2.2 Sentence-Level Event Extraction	19
3.2.3 Document-Level Event Splitting and Linking	21
3.2.4 Attribute Assignment.....	22
3.2.5 Tuning Precision and Recall.....	24
3.3 BBN ACCENT	25
3.3.1 Extended Input and Output Functionality	25
3.3.2 Tools for Data Exploration	26
3.3.3 Robust Error Handling & Recovery	27
3.3.4 Other Robustness Improvements.....	27
3.3.5 Non-Standard Inputs.....	27
3.3.6 Performance Customization	27
3.3.7 Ingest Customization	28
3.3.8 Installation Package.....	29
4 Results and Discussion	30

4.1	Novel Event Class Discovery	30
4.1.1	Automated Evaluation	30
4.1.2	Ablation Experiments.....	31
4.1.3	Manual Evaluation.....	33
4.1.4	Experiments in Next Steps	44
4.2	Civil Unrest	49
4.2.1	Evaluation Data & Metrics	49
4.2.2	Core Evaluation Results.....	53
4.2.3	Secondary Evaluation Results	59
4.3	BBN ACCENT	64
5	Conclusions.....	64
5.1	Novel Event Class Discovery	64
5.2	Civil Unrest	64
5.3	BBN ACCENT	65
6	Recommendations	65
6.1	Novel Event Class Discovery	65
6.2	Civil Unrest	66
6.3	BBN ACCENT	66
7	References	66
	LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS.....	67

List of Figures

Figure 1: Novel Event Discovery Process	2
Figure 2: Overview of Technical Approach	3
Figure 3: Sample of CAMEO XML	26
Figure 4: BBN ACCENT data exploration tool.....	26
Figure 5: ACCENT Parameter Generator.....	29
Figure 6: Sample Screenshot of BBN clustering tool.....	34
Figure 7: Cluster cohesion (size of dominant class)	37
Figure 8: Distribution of pairwise similarity metric output across Accented and Wild candidate pools.....	38
Figure 9: Example of system-generated cluster (Accented condition).....	40
Figure 10: Sample dominant cluster labels	41
Figure 11: Sample of instances from second MakeEmpatheticGesture bucket.....	41
Figure 12: Precision, recall, and F-Measure as tuning parameters vary (sorted by F-Measure) ..	54
Figure 13: Document-Level precision, recall, and F as event mention confidence threshold increases	56
Figure 14: Sentence-Level precision, recall, and F as event mention confidence threshold increases	56

List of Tables

Table 1: Features in similarity metric. Abbreviations are as follows. TG: text graph, WN: WordNet, Source: source actor mention, Target: target actor mention	11
Table 2: Features in event identification. Some features overlap with the similarity metric (reference Table 1). We mark the features used only in event identification with an asterisk (*).	14
Table 3: Classifier performance at different thresholds.....	15
Table 4 - Candidate event anchor <i>a</i> features, grouped by category.....	19
Table 5 - Event anchor <i>a</i> , candidate argument <i>b</i> features, grouped by category	20
Table 6: Pairwise prediction results. Results are averaged across the 4 folds. Note that the F1 presented above is the average F1 across the 4 folds, and not calculated from the averaged recall and precision.	30
Table 7: Automated scoring of the clusters. Results are averaged across the 4 folds.	31
Table 8: Feature ablation results on pairwise predictions. For the reader’s convenience, we also show the results from using all features.....	32
Table 9: Feature ablation results on clustering. For the reader’s convenience, we also show the results from using all features.	32
Table 10: Results when giving different weights to the document vectors.	33
Table 11: Scores for randomly-generated clusters (averaged across both annotators).....	36
Table 12: Scores for randomly-generated clusters in the Wild condition	36
Table 13: Final evaluation scores (cohesion).....	37
Table 14: Final evaluation scores (separation)	37
Table 15: Comparison of two annotators' results on evaluation metrics	42
Table 16: Annotator vs. Annotator agreement.....	43
Table 17: Annotator vs. Annotator/ACCENT agreement for system-generated Accented clusters	43
Table 18: Closest CAMEO classes for clusters discovered by the system.....	44
Table 19: Furthest CAMEO classes for clusters discovered by the system	45

Table 20: Evaluation results for event coding experiment	46
Table 21: Sample attribute correctness	50
Table 22: Overall inter-annotator agreement	51
Table 23: Sentence-level inter-annotator agreement	51
Table 24: Document-level inter-annotator agreement	51
Table 25: Inter-annotator agreement for Attributes	52
Table 26: Overall results	54
Table 27: Sentence-level and document-level Performance	55
Table 28: Human and system performance compared against a single annotator	56
Table 29: System accuracy for attributes	56
Table 30: System accuracy for attributes, as a percentage of human performance	57
Table 31: Comparison of performance using only system output vs. system output seeded with gold sentence-level event mentions and their arguments	59
Table 32: Overall results on Gigaword test set	59
Table 33: Overall results on OSI-positive test set	60
Table 34: Overall Results on Gigaword: Latin America vs. the Middle East (using Gigaword-specific tuning)	61
Table 35: Results on Gigaword: Latin America vs. the Middle East, using models <i>without</i> training data collected for Latin America	61
Table 36: Comparison of performance using fluent English vs. machine translation	62
Table 37: Recall and time saved when skipping documents without system events	64

1 Summary

Forecasting the future is a key part of effective intelligence, and one crucial empowering technology for forecasting models is the automatic extraction of a stream of events (e.g. protests, attacks, etc.) from unstructured text (e.g. news, social media). This technical report presents results from a seedling effort targeting three challenges in the state of the art in event extraction:

Novel event class discovery: Under the Worldwide Integrated Crisis Early Warning System (W-ICEWS) program, BBN extended its state-of-the-art natural language analysis tool BBN SERIF to extract 300 types of socio-political events from text. This new tool (BBN ACCENT) more than doubled the accuracy of the previously deployed open-source solution and is now in regular use at STRATCOM in support of systems monitoring and forecasting national and international crises. However, in real-world deployments, new event classes of interest often arise, whether through changes in the underlying data, in domain, in political climate, or simply in user interest. Meanwhile, existing automatic event extraction systems only code events according to a pre-specified ontology. This technical report describes a proof of concept system developed at BBN that automatically discovers new event types of interest from a stream of (unannotated) text sources, showing a path forward to prolonging the life cycle of an automatic event coding system without requiring systems experts to be called in to make upgrades. We discuss promising results from the proof of concept study as well as lessons learned that could inform broader future work.

Civil unrest: IARPA's Open Source Indicators (OSI) program has successfully developed new approaches to forecasting, focusing on methods for predicting significant societal events from freely-available indicators like social media. Performers are evaluated on the basis of warnings that they deliver about real-world events, including civil unrest, which are compared against "ground truth" events that are manually identified in news reports (after the dates of the predicted events have come and gone). This stream of ground truth events is generated by humans reading news articles and manually coding events. It is used not only to evaluate OSI systems but also later becomes training for forecasting algorithms. Our goal in this effort—using civil unrest as a test case—was to explore whether such ground truth could be created at lower cost and in greater volume for future programs. We discuss our technical approach and promising results.

BBN ACCENT: As mentioned above, BBN ACCENT codes 300 types of socio-political events from text. Until this point, the event coder has been available to a limited set of users. In this effort we extended BBN ACCENT for general release to the research community, adding robust ingest for diverse sources, comprehensive error handling and customization, and other features that will enable effective use across the community.

2 Introduction

2.1 Novel Event Class Discovery

To show evidence that the task of novel event class discovery is feasible, we implemented an experimental framework that takes as input a large unannotated corpus, a database of known actors, an existing event ontology, and an automatic event coder (BBN ACCENT) that identifies events in the existing ontology. Its output is a set of clusters of event candidates (ECs¹), where each EC is a possible event with a **Source actor** and a **Target actor** (e.g. “*Obama met Putin*”), and each cluster of ECs represents a possible event class. A high-level overview of the system is shown in Figure 1:

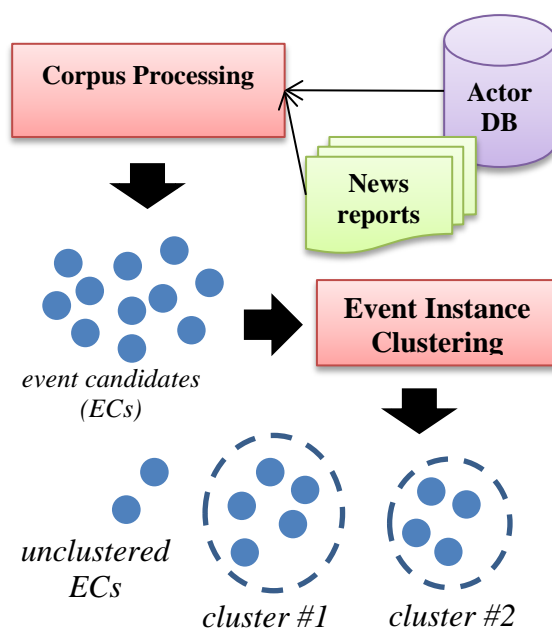


Figure 1: Novel Event Discovery Process

Figure 2 provides more technical details on this process. During training, ECs are extracted from a large, unannotated corpus and are sampled across predicate paths and event codes to provide input to the training process for a trained similarity metric. Event codes produced by BBN ACCENT are used as ground truth for this process. (These codes are drawn from the Conflict and Mediation Event Observations (CAMEO) ontology.) Separately, an event/non-event classifier is trained on ~8000 instances of hand-annotated data created under this effort. During decoding, ECs are first passed through this classifier, which filters out those judged unlikely to be events. The

¹ For a complete list of abbreviations and acronyms referenced in this report, please see page 37.

trained similarity metric then generates pairwise scores for a sample of the remaining instances, which are passed to the clustering algorithm to generate our final clusters, which should ideally represent coherent classes of events.

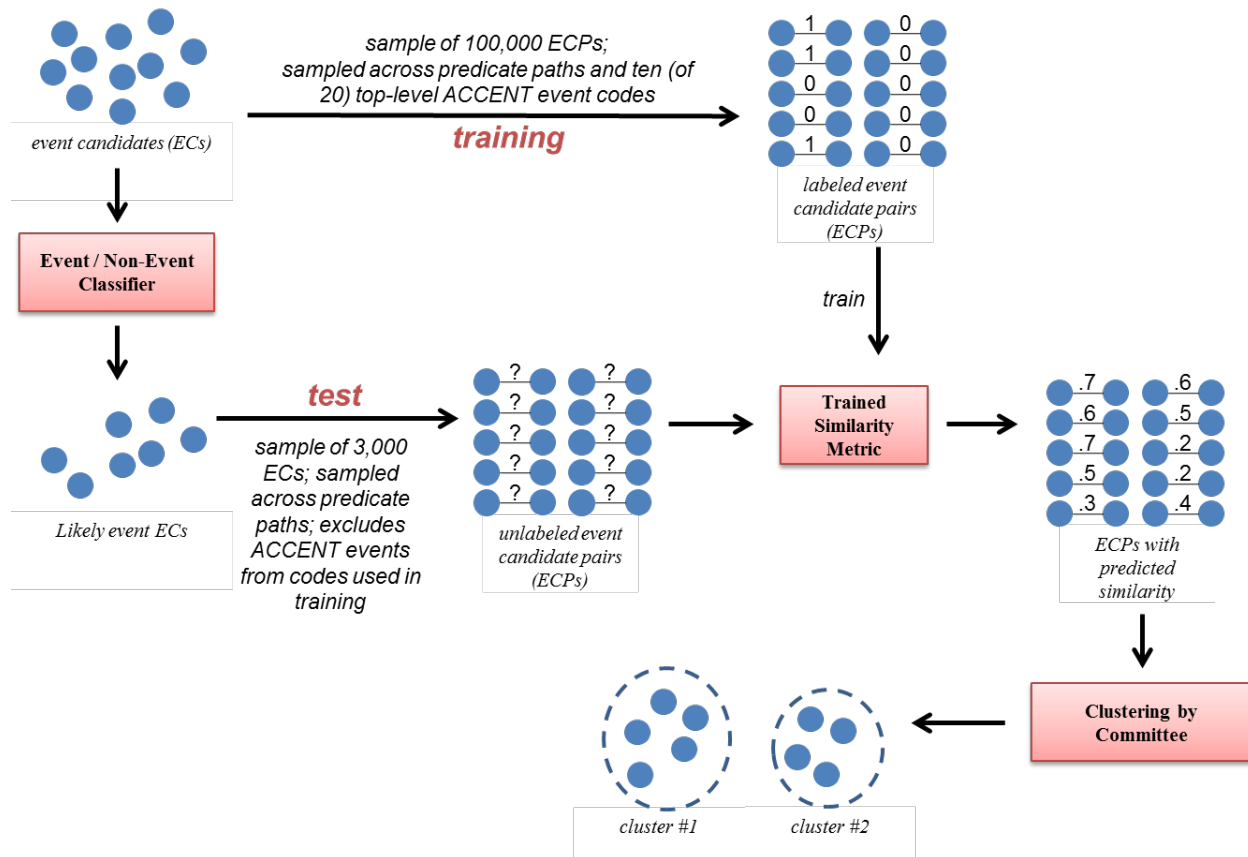


Figure 2: Overview of Technical Approach

We summarize our primary findings below:

- The seedling system does a good job of discovering clusters of similar events when given a pool of CAMEO-like candidates (specifically those ECs judged by BBN ACCENT to be an event in the CAMEO ontology). 86% of evaluated clusters in this condition were judged to be at least moderately cohesive (3+ on a scale of 1-5), and 95% of pairs of clusters were judged to be well-separated (3+ on a scale of 1-5). These results far exceed the evaluation targets in the proposal (40% on each dimension).
- The task is more difficult but the approach still shows promise when given a wider pool of candidates. Here, 39% of evaluated clusters were judged moderately cohesive, and 95% of pairs of clusters were judged to be well-separated. These results exceed or come very close to meeting the evaluation targets in the proposal (40% on each dimension).
- The wider pool is more difficult for several reasons, including:
 - The event/non-event classifier weeds out some but not all of the noise: 10% of candidates are still non-events, even when E/NE classifier is tuned for precision.

- The long-tailed distribution of event classes provides greater challenges in a wider pool. In the limited pool, there were only ~150 classes represented, so a number of event classes had a good number of instances. However, in the wider pool, there seem to be significantly more distinct classes—the pool is dominated by a handful of very common classes and many near-singletons. (This analysis led us to modify our use of the “clustering by committee” algorithm, leading to better results in this condition.)
- This experiment highlighted an interesting and promising side effect of the process: the seedling system often clustered known ACCENT ECs together with ECs that were uncoded by ACCENT. These uncoded events often represented novel phrasings of an event (that ACCENT had previously missed). This suggests that a clustering-based approach to event discovery could help bootstrap recall for known event classes. This is particularly important because recall is a significant known problem for state of the art technology (current best performance is ~30%). Examples also suggest that application of this clustering approach might be particularly promising for recall + domain shift.

2.2 Civil Unrest

BBN’s work in event extraction focuses on two distinct but complementary types of components. Both are built on top of BBN’s core information extraction engine BBN SERIF, which uses statistical models to extract information from text, and both have complementary strengths.

The first component (BBN KBP) involves a set of statistically trained models derived from non-expert human annotation of sample texts. These models identify event “anchors” in the text and attach arguments to them from the nearby context, using both local and document-level features, as well as distributional information derived from large un-annotated corpora. This is the system that is used to participate in the open NIST evaluations.

The second (BBN ACCENT) is a hybrid system designed to leverage human intuition regarding linguistic patterns. To build a model for an event type, a human first decomposes each event type into a combination of smaller building blocks, each of which is represented as a set of text graphs (a BBN-specific representation of predicate-argument structure related to dependency parses, automatically extracted by BBN SERIF at run-time). During model creation, these text graphs are automatically suggested from sample data and then manually expanded and curated to broaden coverage and increase precision. At run-time, BBN ACCENT automatically matches unstructured text to these text graph patterns to produce events with no human participation. Under the W-ICEWS program, BBN ACCENT was extended to automatically extract nearly 300 types of events from open source media in support of political forecasting and analysis, using the Conflict and Mediation Event Observations (CAMEO)² event ontology.

Both systems produce one or more types of event that are relevant to the OSI Civil Unrest category. BBN KBP produces *Conflict.Demonstrate* events (from the community’s ACE ontology) and

² D. Gerner, P. Schrod, Ö. Yilmaz, R. Abu-Jabr. Conflict and Mediation Event Observations (CAMEO): A New Event Data Framework for the Analysis of Foreign Policy Interactions. *Presented at the International Studies Association, New Orleans, and American Political Science Association, Boston (2002).*

BBN ACCENT produces *Protest* events (CAMEO code 14), which are further broken down into subtypes (specifically, *Demonstrate or rally*, *Conduct hunger strike*, *Conduct strike or boycott*, *Obstruct passage*, and *Protest violently / riot*). A second CAMEO event class relevant to this effort is *Coerce* (CAMEO code 17), which represents repressive state actions, including the use of violent tactics to break up protests.

Our work under this effort builds on both BBN KBP and BBN ACCENT to develop an end-to-end system that extracts OSI civil unrest events. In OSI, civil unrest events are defined by the following dimensions:

- The type of population participating in the event, e.g. *Agricultural*
- The date of the event
- The location of the event
- The main objective or reason for the unrest, e.g. *Housing*
- Whether or not the event is violent
- Crowd size

OSI events can have at most one population, date, and location. So, the following sample sentence would contain three events:

In a repeat of last week's protests in New York, teamsters turned out today to rally for higher salaries in both San Francisco and Los Angeles.

- (Labor, last week, New York, Employment and Wages, Nonviolent, ...)
- (Labor, today, San Francisco, Employment and Wages, Nonviolent, ...)
- (Labor, today, Los Angeles, Employment and Wages, Nonviolent, ...)

We summarize our primary findings below:

Overall performance. We evaluated system performance using three tuning settings—one intended to focus on precision, one on recall, and one on F-Measure. On the test set, the more conservative setting (focused on precision) was the most successful, achieving 69% of overall human performance. The balanced setting achieved 64% and the recall setting 59%.

Document-level event detection. When tested simply on its ability to identify which documents contain a civil unrest event coded by at least one annotator, the system achieved a .85 F-Measure. Its recall was .90 and its precision was .81. (The system was not specifically tuned for this task, but it is a helpful diagnostic measure.) The recall and precision balance can be adjusted via a confidence threshold.

Sentence-level event detection. When tested on its ability to identify which sentences contain a civil unrest event coded by at least one annotator (another diagnostic-only measure), the system achieved a .80 F-Measure. Here, recall was .84 and precision .77. As above, the recall and precision balance can be adjusted.

Attribute assignment. Given agreement between the system and an annotator on the existence of an event, the system's average performance on the extraction of the six attributes was just under 90% of human performance. Population, Reason, Violence, and Magnitude are all high (close to or well above 90% of human performance). Date (68%) and Location (74%) posed greater challenges, primarily in terms of inference for dates (e.g. should one assume an undated protest took place on the day the document was written?) and specificity for locations (identifying the country is easier than identifying the city). Notable inter-annotator disagreement on dates (also surrounding appropriate levels of inference) was also likely a source of noise to the trained date attachment models.

Event splitting and linking (event co-reference). The event coding task involves both identifying sentence-level mentions of events in the document as well as appropriately splitting and linking those sentence-level mentions to form real-world document-level event clusters. To show the difficulty of the splitting and linking task, we ran our system with gold sentence-level event mentions provided by the annotators, requiring the system to only perform the splitting and linking task and attribute assignment (which we already assessed separately to be reasonably high). Providing the gold sentence-level event mentions resulted in only a 30% reduction in error, confirming that document-level event analysis poses the most significant challenge and the area most promising for future work.

Precision/recall balance. The confidence measures produced by the system (and used for tuning) proved to be very well-behaved: both precision and recall monotonically increased/decreased (respectively) as the value of the threshold increased. This provides us with the ability to tune the system effectively to different use cases. For instance, if the goal is to find all documents that might have an event, even at the cost of lower precision, we can choose a setting that does so. Similarly if the goal is to identify the subset of events that are most likely to be correct, the system can do that as well.

Inter-annotator agreement. All test documents were coded independently by two trained annotators. Overall agreement was measured at .76 (F-Measure) and document-level agreement was measured at .89 (F-Measure). (Document-level agreement measures only whether the two annotators agreed on whether a document contains at least one codable event.) For four of the six attributes, agreement (given an aligned event) was above .90. Reason and Date both proved more challenging (agreement of .71 and .80 respectively.) For Reason, it appears that the boundaries between categories like *Other Government Policies* and *Other* are simply quite fuzzy. For Date, there was little disagreement when the date of a protest or rally was explicit, but the annotators differed on how aggressive to be when inferring a date for an event from the surrounding context.

Shift in region of interest. The data annotated and evaluated for this effort focused on Latin America, the original area of interest for the OSI program. However, we annotated a supplemental set of data focused on the Middle East, to help determine how much effort would be required to retarget the models to a new region of interest. Initial results showed that our models (trained exclusively on data focused on Latin America) performed much more poorly on the Middle East test data (F-Measure of .54 vs .40). However, we performed a second experiment testing both datasets using an alternative set of models trained using documents annotated prior to this Civil Unrest effort, i.e. data with no bias towards Latin America. The result was only a slight degradation in performance on the Latin America data (0.54 vs. 0.52), but a dramatic improvement on the

Middle East set (from .40 to .51). This experiment seems to indicate that as long as the training data set is sufficiently neutral, transition to a new area of interest with similar performance is not likely to require significant additional work (area-specific annotation might improve performance to some degree, but is not necessary).

Machine translation & native language analysis. We evaluated system performance over both fluent human translations and machine translations (MT) of Spanish newswire. Performance on the two sets was remarkably similar. This runs counter to previous results for BBN ACCENT event extraction over human and machine translations of Arabic newswire, where precision remained relatively stable (a 10% relative decrease), but recall dropped by 50%. We suspect two primary factors allow for better performance over MT for this task: first, the OSI events are reported as normalized attributes rather than actual strings or entities from the text, where MT can prove challenging due to dropped names and garbled entity co-reference. Second, a loss in recall at the event mention level (as measured for BBN ACCENT) may not translate as strongly to a loss in recall at the document level (as for the OSI Civil Unrest task). If the system only finds one of four mentions of an event in a document that results in .25 recall at the sentence level but could still result in perfect recall at the document level.

It is of course possible that performance on fluent English could be more easily improved using features that rely on syntax or other things often garbled by MT. It is also true that this data set is quite small, so conclusions should be held lightly. However, this result, combined with the results above regarding shift in regions of interest, is quite promising for the use of this technology to track events across the globe.

Reduction in level of effort. The ultimate goal of this effort is to explore ways to reduce the level of effort required for the creation of a gold standard. To do this, we measured the time it took to annotate 140 documents both with and without the help of system output as a guide (system events were pre-entered into a GSR-style spreadsheet that annotators were asked to complete). On the one hand, this experiment is artificial—we anticipate that integrated into a real annotation tool, system output could be used much more effectively. In addition, our two trained annotators operate at significantly different speeds, making direct comparison of raw speed difficult. However, we were able to compare a change in the speed ratio between the two annotators when they were given different material to work with. When annotator A was given the raw documents (with no system output), they were 1.96 times as slow as annotator B (who was given the documents with system output). When annotator B was given the documents with the system output and annotator A was given the raw documents, they were only 1.57 times as slow. From this perspective, it appears that providing system output did help speed up the annotation process.

Another dimension we examined arose from the fact that the majority (69%) of these documents had no events (according to the annotators). In most cases, the system correctly did not produce any events for these documents. But in this experiment, the annotator still had to carefully read each of these documents—the system’s opinion that the document contained no events was ignored and irrelevant. We therefore simulated a second experiment where we assumed that annotators skipped all documents in which the system found no events. With the most conservative pruning threshold from our original experiments, this approach resulted in a 44% reduction in time taken while still maintaining a recall of .94. With the most aggressive pruning threshold from our original experiments, the approach resulted in a 62% reduction in time taken while maintaining a recall of

.88. This of course does not reflect any particular effort to maximize performance for this task, so we expect there would be ways to better optimize the system to support this kind of approach. It also does not account for any gains from the actual event mention finding itself (e.g. where in the document the system believes an event to appear, or what its attributes might be)—as mentioned above, this would require integration with a real annotation tool to assess more accurately.

2.3 BBN ACCENT

As mentioned above, BBN ACCENT codes ~300 types of socio-political events from unstructured text (news reports, etc.). It has been evaluated at ~75% precision for the W-ICEWS program, much higher than the previous solution used for W-ICEWS. (Recall performance was also higher than the previous solution but was not measured in absolute terms.) An ACCENT-coded event datastream available for research purpose on a one-year delay through DataVerse, but researchers can't run on their own data (and the long delay might not suit all purposes). The goal of this effort was to make BBN ACCENT more broadly available for research, so that social and political science researchers will be able to develop and test their models using high-accuracy event data from far more sources and domains.

3 Methods, Assumptions, and Procedures

3.1 Novel Event Class Discovery

3.1.1 Event Similarity

A crucial component in effective clustering and discovery of event classes is predicting the similarity/dissimilarity between two event instances. In this section, we first define our learning task and then describe our approach for producing similarity measures.

3.1.1.1 Learning Task

We extract a collection $E = \{EC\}$ of event candidate (EC) instances. Each EC is a possible event with a **Source actor** and a Target actor, such as “*Obama met Putin*”, “*Obama planned to meet Putin*”, or “*Obama’s meeting with Putin*”. Given a pair of ECs, our goal is to determine how similar they are in terms of their *type*³ of event (e.g. meetings, attacks, etc.) To do this, we define each instance x in our learning formulation as $x = (a, b)$, where $a \in E$, and $b \in E$. Each instance x is associated with a true label $y \in \{0,1\}$, which takes the value 1 if a and b are in the same event class, and 0 otherwise. We then use these binary labels to train models from which we can derive a similarity measure (something not present directly in the training data).

We now describe the details of our event similarity measures and features.

³ Note that we are not performing event coreference.

3.1.1.2 Same/Different Event Class Classifier

Given two candidate event instances (a, b) , for instance $a = “M_i \text{ planned to criticize } M_j”$ and $b = “M_k \text{ admonished } M_l”$, we train a logistic regression classifier that predicts whether a and b are in the same or different event class. Logistic regression is used because it provides well-calibrated probability estimates of the target classes; we experimented with other classifiers such as a linear SVM, but we found that performed substantially worse in cases where the positive/negative example balance was far from 50/50, as it is in our application. The classifier produces the predicated probability that (a, b) are in the same event class, which we then feed into our downstream clustering algorithms. However, this approach did not perform as well as the second (the cosine similarity optimizer, described below) in the intermediate evaluation, so we did not pursue it past that point. (Discussion of why cosine similarity might be a better fit can be found in in Section 4.1.1.1, which presents the evaluation results for the similarity metric.)

3.1.1.3 Similarity Measure Optimizer

Cosine similarity is a popular similarity measure. Given two ECs a and b , where each is a real vector of m dimensions, the cosine similarity is defined as follows:

$$\cos(a, b) = \frac{\sum_{k=1}^m a_k b_k}{\sqrt{\sum_{k=1}^m a_k^2} \sqrt{\sum_{k=1}^m b_k^2}}$$

In our work here, we parameterize cosine similarity with a m dimensional weight vector $\theta = (w_1, \dots, w_k, \dots, w_m)$:

$$\cos_{\theta}(a, b) = \frac{\sum_{k=1}^m w_k a_k b_k}{\sqrt{\sum_{k=1}^m w_k a_k^2} \sqrt{\sum_{k=1}^m w_k b_k^2}} \quad (1)$$

To learn the weight vector θ , we use gradient descent with log-loss⁴ as our cost function. The range of $\cos_{\theta}(a, b)$ is $[0.0 - 1.0]$, which we use as our main measure of similarity between two ECs.

3.1.1.4 Features

We explored multiple features for predicting event similarity. For instance, we extract the ICEWS sectors (e.g. Government, Military, Rebel, etc.) associated with each source and target actor mention in the actor dictionary. These sectors serve as general groupings of actor mentions, and we theorize that similar groups of actor mentions perform (or are the target of) similar event types.

Given an EC such as “ M_i planned to criticize M_j ” (where M_i and M_j are the respective Source and Target ICEWS actor mentions), we also extract its associated text graph (TG) “ $M_i \langle sub \rangle$ planned $\langle to \rangle$ criticize $\langle obj \rangle M_j$ ”, where angle brackets $\langle \cdot \rangle$ represent proposition *role* labels (similar to dependency roles) between *predicate* words such as “planned” and “criticize”. Features derived

⁴ To ensure that the log-loss is computable, $\cos_{\theta}(a, b)$ must return a positive value. In practice, we ensure this by constraining all model features and weights to positive values.

from text graphs (using similarity of both roles and predicates) build on previous similarity detection work at BBN. Note that our features do *not* include actual predicate words (e.g. *predicate=arrest*), since the similarity metric needs to generalize across classes. Instead all predicate-based features are similarity-based, e.g. *shares-predicate*.

We also employ word embeddings to represent *predicates* in this work. Word embeddings can help calculate indirect synonymy (rather than the direct kind of synonymy represented in something like WordNet). In our mid-term evaluation, we used the off-the-shelf embedding vectors from Baroni et al. (2014) which was trained from the English Wikipedia and UkWac (UK web corpus). However, since our evaluation focus is on the English Gigaword, we trained word embeddings (Mikolov et al., 2013) on all of English Gigaword 5. As a filtering step, we restrict our word vocabulary to words containing at least 2 alphabet letters and appearing at least 10 times in the Gigaword corpus. We then trained using the CBOW architecture with negative sampling. As hyper-parameters, we use 2 training iterations, 10 negative samples, context window size of 5, and 400 hidden units. When training concludes, each word is represented by a vector of 400 real numbers.

It also seemed useful to examine broader document context. Topic detection (e.g. via LDA) is a natural fit for a document similarity feature; however, Le and Mikolov (2014) recently showed an embedding-based approach is even better. So, following their approach, we trained paragraph vectors on the same English Gigaword corpus and parameters as above. Since paragraphs are not defined in the English Gigaword, for simplicity, we define paragraphs as documents. When training concludes, each Gigaword document is represented by a vector of 400 real numbers.

We summarize our features in Table 1 and describe them below. Note that unlike the usual classification task based on individual instances or examples, our learning task is based on pairs of examples, or pairs of ECs. Hence, as an example feature type “share WN synset”, we check whether any predicate pair (one predicate from each EC) shares the same WN synset.

Table 1: Features in similarity metric. Abbreviations are as follows. TG: text graph, WN: WordNet, Source: source actor mention, Target: target actor mention

Feature category	Feature type	Description
Predicates in TG	Word embeddings	A pair of dense embedding vectors representing the most similar predicate pair.
	WN similarity	A number giving the max WN-based similarity score of predicate pairs.
	Share WN synset	Whether any predicate pair shares the same WN synset.
	Share word	Whether any predicate pair is the same word.
	Share stem	Whether any predicate pair is the same stem.
Proposition roles in TG	Direct roles	The proposition role of the source and target.
	Roles on path	The set of proposition roles on the TG path from source to target.
	Role sequence patterns	Generalizations of the sequence of proposition roles from source to target.
	TG structure substitution	Learn substitution costs using role pairs, and role-word pairs.
Argument	Sector overlap	Numbers representing sector similarity between source mentions and sector similarity between target mentions.
	Neighboring entity types	Features (role, entity-type) of the words directly connected (via propositions) to the source or target mentions.
Document level	Document vectors	A pair of dense embedding vectors representing the associated document pairs.

Using the predicates from the associated text graphs (TG) of ECs, we extract the following features:

- **Word embeddings:** Given two ECs (a, b) with associated predicate words $P_a = \{\text{planned, criticize}\}$ and $P_b = \{\text{admonished}\}$, we first obtain all pairs of predicates $P_a \times P_b$. For each $(p_a, p_b) \in P_a \times P_b$, we represent each predicate p with its dense embedding vector, and calculate the (unparameterized) cosine similarity $\cos(p_a, p_b)$. The (p_a, p_b) pair with the highest cosine similarity is chosen (e.g. $p_a = \text{criticize}$ and $p_b = \text{admonished}$) and their associated embeddings are used as features.
- **WordNet similarity:** We compute a WordNet-based similarity metric score (Wu and Palmer, 1994), which provides a measure of how similar two words are. This measure reflects the relationship between the stems of the two words in the WordNet synset hierarchy and how specific or broad the most-specific concept that includes both words is.
- **Share WordNet synset:** For each predicate pair $(p_a, p_b) \in P_a \times P_b$, we compute whether there exists any synset (group of synonymous words) that p_a and p_b are both a member of, i.e. $\exists (p_a, p_b) \in P_a \times P_b$, where $\text{synsets}(\text{stem}(p_a)) \cap \text{synsets}(\text{stem}(p_b)) \neq \emptyset$.

- **Share predicate word:** Similar to the above, this is a binary feature. We compute whether there is any predicate pair (p_a, p_b) , where $p_a == p_b$ (i.e. the two predicates are the same word).
- **Share predicate stem:** This is similar to the above, but we compare predicate pairs after stemming them.

Leveraging the proposition roles from the TGs, we extract the following features:

- **Direct proposition roles:** We extract the proposition roles directly connected to the source and target actor mentions. For example, given “Obama $\langle sub \rangle$ planned $\langle to \rangle$ meet $\langle obj \rangle$ Putin”, we extract $\langle sub \rangle$ for source and $\langle obj \rangle$ for target.
- **Proposition roles on path:** The set of roles on the proposition path from source to target actor mentions. For instance, $\{\langle sub \rangle, \langle to \rangle, \langle obj \rangle\}$
- **Proposition roles sequence pattern:** We extract the sequence of proposition roles from source to target mention, but optionally omit roles on the path for generalization. For example, given the text graph “Obama $\langle sub \rangle$ planned $\langle to \rangle$ meet $\langle obj \rangle$ Putin” and assuming *meet* is the root of the text graph, we extract the following features:
 - From source to root: $\langle sub \rangle$ -, *, *- $\langle to \rangle$
 - From source to target: $\langle sub \rangle$ -, $\langle to \rangle$ -, *, $\langle sub \rangle$ -, *- $\langle obj \rangle$, *- $\langle to \rangle$ -, $\langle obj \rangle$
 - There is no target to root specific features since we require a minimum of 2 proposition roles on the sequence, and there is only a single $\langle obj \rangle$ role from target to root.
- **Text graph structure:** To estimate the distance between phrases such as P1: “X’s objection against Y” and P2: “X protested Y”, we leverage their associated text graphs: “X $\langle poss \rangle$ objection $\langle against \rangle$ Y”, and “X $\langle sub \rangle$ protested $\langle obj \rangle$ Y”. For instance, if we know that substituting a $\langle sub \rangle$ proposition role for a $\langle poss \rangle$ role is a low cost operation, this helps to determine that phrases P1 and P2 are similar. Contrast this with P3: “Y protested X” with associated text graph “X $\langle obj \rangle$ protested $\langle sub \rangle$ Y”. Knowing that substituting $\langle sub \rangle$ for $\langle obj \rangle$ is high cost would help determine that P2 is dissimilar to P3. In this work, we automatically learn such substitution costs on roles and predicates. To do this, we extract the following types of features:
 - Role-substitution: We first extract the set L of proposition role labels (or sequence of role labels) from the *source* actor mention to the root of the text graph. For example, $L_{P1} = \{\langle poss \rangle\}$ (assuming *objection* is the proposition root), and $L_{P2} = \{\langle sub \rangle\}$ (assuming *protested* is the proposition root). We then use label pairs $(l_{P1}, l_{P2}) \in L_{P1} \times L_{P2}$ as features. In this example, this is just the single pair $(\langle poss \rangle, \langle sub \rangle)$.
 - Weighted-role-substitution: Similar to role-substitution, but we weigh each label pair by the similarity of their ancestor words. In our example, we will weigh the label pair $(\langle poss \rangle, \langle sub \rangle)$ by the cosine similarity between the word pairs $(objection, protested)$, where each word is represented by their word embeddings.
 - We also extract similar features from the *target* actor mention to the root of the text graph.

We also extract information from the source and target actor mentions, and their surrounding context:

- **Sector overlap:** The ICEWS sectors (e.g. Government, Military, Rebel, etc.) associated with each actor mention serve as general groupings of actor mentions. We extract the following features:
 - Source-sector-overlap: Given that M_i and M_k are the respective *source* actor mentions in event instances (a, b) , we first get all of their respective ICEWS sectors S_i and S_k . We then calculate the associated Jaccard index $\frac{|S_i \cap S_k|}{|S_i \cup S_k|}$ as a measure of their sector(s) similarity.
 - Target-sector-overlap: This is similar to the above, except that we calculate over ICEWS sectors S_j and S_l associated with the *target* actor mentions.
- **Neighboring entity types:** This captures the proposition structural context of the source and target mentions. For each word w that is directly connected to the source (or target) mention via a proposition role label l , we extract a feature pair (l, ET_w) , where ET_w is the predicted entity type (e.g. PER, ORG, GPE, etc.) of w .

We did not include features based on date and location as given our observation of the data, they did not seem likely to improve performance; we focused our efforts on features that seemed more likely to impact the final results.

We learn the weights of the above features using the parameterized similarity measure $\cos_\theta(a, b)$ defined in Equation 1 (page 9), which captures the intra-sentence similarity between ECs a and b .

3.1.1.5 Decoding

To include document-level context when measuring similarity between a pair of ECs a and b , we calculate the cosine similarity $\cos_D(a, b)$ between the document (embedding) vectors associated with the documents from which a and b were drawn. Our final event similarity score between a and b is then:

$$\text{sim}(a, b) = \alpha \cos_\theta(a, b) + (1 - \alpha) \cos_D(a, b) \quad (2)$$

Using experiments on development data, we set $\alpha = 0.95$. The final similarity score $\text{sim}(a, b)$ is a real number $[0.0 - 1.0]$, which we subsequently feed to a clustering algorithm. In our work, we use the Clustering By Committee algorithm (CBC, Pantel and Lin 2002) to form clusters that represent event classes.

3.1.2 Event/Non-Event Classification

Given the sentence “Under a *Haiti* - *U.S.* agreement, would-be *immigrants* are returned once they are intercepted.”, there is an (agreement) event between the actor mentions “*Haiti*” and “*U.S.*”, but an event is not explicitly attested between “*Haiti*” and “*immigrants*”. In particular, a large portion of intra-sentence EC pairs do not form a valid event example. Eliminating these examples prior to clustering should enable forming more cohesive event clusters. Thus, we develop a binary classifier to identify EC pairs that are likely to be events.

For our purposes, we defined an event as follows:

- Actors are specific persons, organizations or locations
- Connection between actors can be succinctly generalized (e.g., *Source gives deadline to Target*)
- Actors have a direct connection to each other; they are not merely both acted on by a third party
- Connection conveys more than just location/membership

We annotated 8000+ ECs as event or non-event for this task; a small portion were double-annotated, showing an agreement of ~80%.

For our experiments, we trained a logistic regression binary classifier using the implementation in Liblinear (Fan et al, 2008). Note that unlike the similarity metric which operates on EC pairs, event identification here operates on individual EC instances. We summarize our features in Table 2. Some of these are also used in the similarity metric (reference Table 1). We describe the features that are only used in event identification below.

Table 2: Features in event identification. Some features overlap with the similarity metric (reference Table 1). We mark the features used only in event identification with an asterisk (*).

Feature category	Feature type	Description
Predicates in TG	*Words on path	The set of non-auxiliary verb predicates (stem) on the TG path from source to target.
Proposition roles in TG	Direct roles	The proposition role of the source and target .
	Roles on path	The set of proposition roles on the TG path from source to target.
	Role sequence patterns	Generalizations of the sequence of proposition roles from source to target.
	*Role sequence	The sequence of proposition roles from source to target (but excluding the source and target direct roles)
Argument	Neighbor entity types	Features (role, entity-type) of the words directly connected (via propositions) to the source or target mentions.
	*Neighbor words	Features (role, word) of the words directly connected (via propositions) to the source or target mentions.
	*Sector pairs	Sector pairs of source and target mentions.

- **Words on path:** From the text graph (TG) associated with the event candidate (EC), we extract the set of non-auxiliary verb predicates on the proposition path from the source to target actor mention.
- **Role sequence:** The sequence of proposition role(s) from the source to target mention. We exclude the direct proposition roles since they are already covered by a separate feature.
- **Neighbor words:** This captures the proposition structural context of the source and target mentions. For each word w that is directly connected to the source (or target) mention via a proposition role label l , we extract a feature pair (l, w) .

- **Sector pairs:** Given an EC with a source s and target t actor mention, let S_s and S_t denote the respective sets of ICEWS sectors associated with these mentions. We extract sector pairs $(s_s, s_t) \in S_s \times S_t$ as features.

Given a trained classifier, the second question is the optimal operating point of that classifier. The binary logistic regression classifier (based on the Sigmoid function) is inherently trained to maximize classification accuracy using the posterior probability thresholded at 0.5. But in our case, this tilts predictions towards recall, which is not ideal for our purposes. We decided to select for precision: less than perfect recall is acceptable, since our goal is not necessarily to cluster the whole space, but to find meaningful clusters of new events, and non-event noise seems to significantly hinder that goal. A precision of 80% seemed like a reasonable target, so we selected a classifier threshold of 0.75. The following table shows performance across different classifier thresholds:

Table 3: Classifier performance at different thresholds

Threshold \rightarrow	0.5	0.575	0.75	0.8
Accuracy	0.69	0.69	0.64	0.59
Precision	0.71	0.73	0.80	0.83
Recall	0.80	0.73	0.49	0.35
F	0.75	0.73	0.61	0.50

3.1.3 Clustering

In this section, we describe the Clustering By Committee algorithm (CBC, Pantel and Lin 2002) algorithm. CBC was originally developed to discover and cluster word senses and has been wildly popular since its introduction, gathering approximately 700 citations to date. The overall approach is as follows (asterisk * indicate hyper-parameters in CBC):

Input: given a set N of ECs to cluster:

1. For each EC e_i , locate its top* most similar ECs T_{e_i} from the other $|N - 1|$ ECs
2. Cluster T_{e_i} using hierarchical agglomerative clustering (HAC). Select the most cohesive cluster c_{e_i} from HAC.
3. We call c_{e_i} a “committee”, and we have N committees, one for each e_i . Rank these committees in decreasing order of cohesion scores.
4. Going through the ranked list, iteratively add committees, if they are not similar* to previously added committees.
5. For all committees $\{c_{e_i}\}$ that are not added, collect the residue set of ECs $R = \{e_i\}$.
6. Set $N' = \emptyset$. For each residue EC $e_i \in R$, add it to the set N' if it is not similar* to any of the added committees.
7. If there is no new committees added in this iteration, or no residue EC added to N' , stop. Else, set $N = N'$ and repeat by going back to the first step.

Output: A set of committees

CBC is suited for our task of event discovery for several reasons:

- First, CBC automatically discovers the number of committees, and hence the number of event classes from data, based on its hyper-parameters (asterisk * in Figure 1). This is an advantage over clustering algorithms such as K-means where one has to explicitly specify the number of clusters. CBC is also deterministic where each run on the same target corpus produces the same committee clusters. Contrast this with K-means where the initial centroids are randomly selected from run to run.
- Second, the committee clusters produced by CBC do not attempt to cover the entire space of input ECs. In our experiments on the ‘wild’ condition where we ran CBC over 3,000 ECs, approximately half of those ECs are members of committees. The remaining half are left un-clustered, presumably due to noise (e.g. they might be non-events) or they belong in the long tail of near singleton event classes where it is impractical to form cohesive event clusters.
- Third, we found that in our experiments, a committee EC is involved in an average of 5 committee clusters. These might represent different granularities of event classes, opening up opportunities for future work in this direction.
- Finally, CBC is a “two-in-one” clustering algorithm. It internally leverages the hierarchical agglomerative clustering (HAC) algorithm to produce cohesive committee clusters.

When CBC concludes, due to the nature of steps 2 and 4 in the algorithm, it produces a set of cohesive committees which are far apart from one another in the similarity space. We can then either use the committees directly as event clusters or we can re-examine the whole space and cluster ECs together according to the committee to which they are most similar. Our initial approach was the latter; this is what is reported in the Accented condition. However, we explored both options in the Wild condition after discovering that the use of committees provided better results.

Rather than explore multiple clustering methods or combinations of clustering methods, we decided to focus on this single clustering method that seemed well-suited to our task. (The CBC approach in itself, though, does involve multiple approaches to clustering in its two-step approach, as mentioned above.) This was an element of the original proposal that we might have explored in greater depth had the event/non-event classifier not emerged as a necessary addition to the effort; on the other hand, as our work progressed, we also judged exploration of additional features for the similarity metric more likely to bear fruit. Exploration of other clustering methods could be an interesting part of future work. We also had considered exploring clustering on subsets of the original data constrained by actor or date, but as we worked further with the data, this did not appear likely to yield highly productive results.

3.1.4 Experiment Data

We use documents from the year 2002 of the English Gigaword corpus version 5 as development data, and year 2003 as evaluation data. We apply BBN ACCENT and the ICEWS actor dictionary to identify ICEWS actor mentions and ACCENT event instances according to the CAMEO ontology.

We generate our experiment data using the following three criteria:

- We define an event candidate (EC) as a pair of ICEWS actor mentions connected by SERIF propositions. We limit the distance of the relationship between the actors to four *hops*, where a hop is defined as an edge in the proposition graph connecting the actors. For example, “*Obama met Putin*” is considered a two-hop event ($Obama \rightarrow met, met \rightarrow Putin$), “*Obama planned to meet Putin*” is a three-hop event ($Obama \rightarrow planned, planned \rightarrow meet, meet \rightarrow Putin$), and “*Obama considered planning to meet Putin*” is a four-hop event. 97% of ACCENT-coded CAMEO events have three hops or fewer; about two-thirds of all syntactically-connected actor pairs have three hops or fewer. (Note that the distribution over ACCENT events is not necessarily a representative distribution of the true space of events, as BBN ACCENT’s models are biased towards actor pairs that are more closely connected syntactically. However, the distribution over all actor pairs is also not necessarily a representative distribution of the true space of events, as pairs that are more syntactically distant are typically less likely to be eventlike. Further work could explore actor pairs that are more syntactically distant)
- It is uninteresting if our learners simply learn to cluster ECs together based on them having the exact same predicate words. Instead, we would like our learners to predict that “M1 criticized M2” and “M3 reprimanded M4” are similar events. Thus, we ensure diversity in our examples by only considering examples with unique proposition/predicate connections. This uniqueness requirement is defined over the entire series of predicate stems, so “Obama criticized Putin” and “Obama planned to criticize Putin” are considered unique predicate connections, while “Obama criticized Putin”, “Obama criticizes Putin”, and “Obama criticized Bush” are all considered to be the same. This makes our prediction task harder, but potentially more interesting. (In a simple experiment during the intermediate evaluation, removing this constraint improved our scores by 20% relative.)
- We randomly split the CAMEO event codes into two approximately equal portions, one to generate training examples, and the other to generate development and test examples. This ensures that we evaluate on novel event codes, testing the capability of our learners to generalize to codes unseen in training. This effectively tests a scenario where we begin with only a subset of the CAMEO ontology (the ones used in training) and are testing the system’s ability to discover the classes represented by the rest of the ontology, as well as novel classes.

In our experiments, we generate our data as follows. We first extract all ECs from the Gigaword texts, while adhering to our above criteria of maximum hop distance, and predicate diversity. These ECs are then feed to our event identification system for prediction. Using only the ECs that are predicted to be valid events, we then generate training, development, and test data using the processes described below.

3.1.4.1 Training and Development Data

Here, we aim to generate event-pair examples $X = \{x_1, \dots, x_i, \dots, x_N\}$, where x_i represents a pair of ACCENT-coded event instances (a, b) , and has an associated label $y_i \in \{0, 1\}$. We target a 10%/90% distribution of positive/negative examples (10% same class, 90% different class) while generating this dataset. This distribution approximates the natural distribution in ACCENT’s output. We will use these examples to train and develop the similarity metric algorithm.

To generate a set X of examples, we use the following sampling process:

1. Randomly select 2 CAMEO event codes c_1 and c_2 . Sampling across event codes helps to ensure that our resulting X examples contain a more uniform distribution across all event codes than a natural sample by reducing the bias toward the most frequent codes. This aids training and development by maximizing the information density of the training data. This pseudo-stratification across event codes is only applied to training and development data, not test data.
2. If the set does not already contain the desired number of positive examples, randomly draw an event instance pair (a_{c_1}, b_{c_1}) from the set of ACCENT c_1 coded instances. This forms a positive (label $y=1$) example. We similarly draw another positive example (a_{c_2}, b_{c_2}) .
3. If the set does not already contain the desired number of negative examples, randomly draw 2 negative examples, i.e. instance pairs (a_{c_1}, b_{c_2}) and (a_{c_2}, b_{c_1}) .
4. Repeat the above 3 steps until we obtain the desired number of total examples

We use the above process to generate **Training** and **Development** data, with each dataset having $N=100K$ examples. We train the similarity measure optimizer on the training data, while tuning them on the development data.

3.1.4.2 Test Data

We aim to evaluate the performance of our learning algorithms in two settings: (i) over ACCENT-coded event instances, (ii) over all candidate event instances (where the majority will not be ACCENT-coded). We subsequently generated the following three different datasets for evaluation. As noted above, these data sets are subject to the limit on number of hops between source and target actors, the enforcement of diversity across predicates (not producing multiple examples of “X met Y”), and the filtering of any ACCENT-coded events such that the event codes used in training do not appear in the test set.

ACCENTED: We randomly selected a set K of ACCENT-coded event instances. In our experiments, we set $|K|=3,000$. We then form all possible K -choose-2 combinations of instance-pairs (~ 4.5 million instance-pairs) for prediction and clustering.

WILD: Unlike the above dataset which restricts to just ACCENT-coded instances, here we consider *all* ECs, and randomly select a set of 3,000 instances. We similarly form all K -choose-2 combinations for prediction and clustering. In our experiments, we observe that only a very small portion of these instances (just 6%) are ACCENT-coded instances. There are two primary reasons for this. First, the CAMEO event taxonomy, though large, does not cover all possible event types. This is expected; expansion beyond CAMEO motivates this entire effort. Second, ACCENT misses some number of true CAMEO events. And finally, a large number of actor mention pairs are simply not involved in any event together (many but not all of these are removed by the event/non-event classifier).

3.2 Civil Unrest

3.2.1 Overview

The overall operation of the BBN Civil Unrest system is as follows:

1. Process the input data using the core BBN SERIF natural language processing suite.
2. Extract sentence-level civil unrest events using both BBN KBP and BBN ACCENT.
3. Combine sentence-level event mentions into document-level event clusters. This involves splitting some event mentions (e.g. those with multiple Location arguments) and combining others (e.g. those referring to the same real-world event but in different sentences).
4. Assign attributes to document-level event clusters.

We describe each of these components in detail below.

3.2.2 Sentence-Level Event Extraction

3.2.2.1 *BBN KBP*

BBN SERIF applies a pipeline of natural language processing (NLP) analytics to unstructured text, extracting information such as parse trees, entities, relations, and events. BBN SERIF also extracts text graphs (TGs), which are enriched dependency structures automatically built from parse trees. Text graphs differ from other dependency representations in that they have more extensive normalization and incorporate long-distance dependencies.

BBN KBP performs sentence level event extraction using the output of BBN SERIF as input to two models: one for anchor identification and one for argument attachment. The anchor identification model is a supervised logistic regression model which marks words as anchoring a civil unrest event (or not). Table 4 shows the different features used by the argument identification model. The contextual embedding features use dense vector representations for the context of the candidate anchor in the surrounding text graph.

Table 4 - Candidate event anchor a features, grouped by category

Category	Feature
BBN ACCENT	Event types found by BBN ACCENT in the document containing a Event types found by BBN ACCENT in the sentence containing a
Clusters	Brown cluster bit strings of a (at bit lengths 8, 12, 16, 20)
Contextual embeddings	Preposition, adverb, or particle following a
N-grams	a , word before a , word after a , and combinations thereof Stemmed forms of n-grams
Noun Compounds	If a is part of a noun compound, a 's relative position in the noun compound
TG - General	Subject, object, and other children of a Subject, object, and other children of a in conjunction with a Subject, object, and other children of a in conjunction with a 's cluster bit strings
TG - Locations	Role of any geopolitical entity, facility, or location connected to a Whether there is a locative modifier to a
Topic	Document topic Document topic in conjunction with a Document topic in conjunction with a 's cluster bit strings
Word Class	Presence of a on a list of known words associated with event
WordNet	Hypernym synsets of a

The argument attachment model is a supervised logistic regression model which relies on the anchor model described above to identify event anchors. Given an anchor-argument pair (a, b) , the model predicts whether any of the predefined event argument roles hold between (a, b) . We list our event argument features in Table 5.

Table 5 - Event anchor a , candidate argument b features, grouped by category

Category	Feature
Argument	Headword of b Headword of b in conjunction with a
Distance	Number of tokens between a and b in conjunction with b 's entity type
Intervening Text	Words between a and b Stemmed words between a and b Words between a and b , excluding some words based on their part of speech
Text Graph	Role and entity type of b The sequence of text graph edge labels connecting a to b The unordered set of text graph edge labels connecting a to b The role of b if there is a path connecting a and b The set of all (role, word) pairs for words directly connected to b

3.2.2.2 BBN ACCENT

To supplement the BBN KBP sentence-level event extraction model, we also make use of BBN ACCENT's event coding output. BBN ACCENT is a state-of-the-art event coding system that extracts events from text in accordance with the Conflict and Mediation Event Observations (CAMEO) ontology.

The CAMEO ontology requires all events to have both a Source and Target specified, in the same sentence. Under separate funding, we expanded BBN ACCENT to code "monadic" protests,

specifically those where no Target is explicitly specified (or at least not specified in the same sentence). For this effort, we further expanded the system in a limited way to extract at least a subset of Protest events with *no* Source or Target argument specified, e.g. in the sentence “*a protest rally was held today*”. To do this, we took the text graphs in the binary and monadic models and removed the nodes that required Source or Target actors. We then tested the results and removed the pruned text graphs which produced too many spurious events. We also added a small number of additional high-yield no-argument text graphs based on observation of the data. However, we do not consider this new model “complete”, but rather a baseline for no-argument events.

We used the output of BBN ACCENT as both a feature in the BBN KBP sentence-level models and also as the basis of our Violence detection component (see Section 3.2.4.3).

3.2.3 Document-Level Event Splitting and Linking

From the sentence-level event mentions produced by the system, we create document level events (*event clusters*). We achieve this in two steps. First, we split each sentence-level event mention into one or more event-mentions based on Location and Date arguments. So, an event mention with two distinct Locations and one distinct Date will be split into two event mentions, each inheriting one distinct Location from the original event mention along with the only Date argument. Similarly, an event mention with two distinct Locations and two distinct Dates will be split into a total of four event mentions each inheriting one distinct Location and one distinct Date from the original event mention. This is consistent with the definition of an event for the OSI program.

Two Location arguments are considered distinct if both of them can be resolved to a location in the gazetteer, the resolved locations are not the same, and neither subsumes the other one. For example, if a place argument is resolved to “*La Plata, Buenos Aires, Argentina*” and another is resolved to “*Miramar, Buenos Aires, Argentina*”, they are distinct. However, if one argument is resolved to “*La Plata, Buenos Aires, Argentina*” and the other could only be resolved up to the province/state level to, say, “*_, Buenos Aires, Argentina*”, they are not distinct, since the latter subsumes the former. If a Location argument cannot be resolved to any level (city, state or country), it is ignored during splitting.

Similarly, two Date arguments are considered distinct if both of them can be resolved to a specific time or time-period, the resolved times or time-periods are not the same, and neither one subsumes the other one. For example, “*2012-03-09*” and “*2012-03-10*” are distinct, but “*2012-03-09*” and “*2012-03*” are not. If a time argument cannot be resolved to a specific time period (year, month of year, week of year, day of month, etc.), it is ignored during splitting.

After sentence-level event mentions are split on Date and Location, we cluster or link the event mentions based on Date and Location arguments. We merge two event mentions if and only if there is no conflict between their Date and Location arguments (using the guidelines above to define conflict). Additionally, we apply a sentence-distance constraint on the event mentions to be clustered. Using this constraint, we link two event mentions only if the minimum sentence distance between the closest anchors of the two event mentions is within a certain limit.

3.2.4 Attribute Assignment

3.2.4.1 Population

The process for assigning Population attributes relies primarily on the output of the sentence-level event extraction models, which identify the mentions of entities (persons, organizations, etc.) who are participants in civil unrest events.

First, as a pre-process, all entity mentions in the document are classified by population type. This process is done during regular BBN SERIF processing using a set of patterns derived from the OSI guidelines and the training data (e.g. *farmers* are Agricultural, any entity modified by the word *hospital* is Medical, etc.). The patterns are written in the BBN SERIF pattern language, a flexible and powerful representation of syntactic constraints that make these patterns easy and efficient to construct.

The process then proceeds as follows:

- Examine all entity mentions extracted as event participants by the sentence-level event extraction models.
 - If any has an explicit population type, select that type.
 - If any is in an employment, membership, or subsidiary relation with an explicit population type, select that type. (These relations are provided by the standard BBN SERIF analysis.)
- If nothing is selected above, examine all entity mentions that are co-referent across the document with the entity mentions above. Follow the same guidelines to select an explicit population type, if possible.
- If nothing is selected above, apply a small set of phrase-based heuristics to select certain types of populations that are typically missed by the above processes. Phrases were selected from the training data; their presence in the sentence allows a certain population type to be selected. Examples of such phrases are *school protest* (indicating Education) and *walkout* (indicating Labor).
- Finally, if nothing has been selected above, but the Reason for the protest has been identified as Employment & Wages, classify the Population as Labor.

3.2.4.2 Reason

Reason detection is a challenging process. The agreement between annotators is only .72, the lowest of all six attributes. In addition, most of the reason classes are quite rare. Most things are classified as *Other Government Policies*, or sometimes just *Other*—but the agreement on that distinction is particularly low even among humans. We explored training a model for this attribute: we had annotators mark the specific spans conveying their rationale for selecting a particular Reason, and used that information as well as general context as features to the model. However, the model underperformed a simpler approach that simply used a list of key phrases (derived from the spans marked in the training data) to indicate the less-frequent types of protests, and classified everything else as *Other Government Policies*.

3.2.4.3 Violence

We experimented with two approaches to assigning Violence attributes to our event clusters. In one approach, we trained a model using our Civil Unrest annotation data. The model’s task was to predict, for each event cluster, whether the event was violent or not. One challenge with this approach was that the annotation did not provide any information about where the indicators of violence are. So, an event might have an anchor in seven sentences in a document, but only one sentence conveys evidence of violence. When the model attempts to learn how Violence is expressed from all seven sentences (despite violence not being present in six), a significant amount of noise is introduced into the process.

Our other, more successful, approach was to leverage the already-existing violent/non-violent distinction present in CAMEO. If a sentence in which an event mention is found also contains a CAMEO 145 event (*Protest violently*), the event mention is marked as Violent. We also mark as violent any event mention in a sentence containing an act of violent repression by law enforcement (CAMEO 175). Violence attributes then propagate to event clusters (if at least one event mention is marked as Violent, the whole cluster is marked as Violent).

The primary sources of error with this second approach involved the no-argument events mentioned in Section 3.2.2.2—specifically that BBN ACCENT does not typically find no-argument events, and its expansion to do so for this effort is not comprehensive. So, some violent protest events are missed because BBN ACCENT fails to find a CAMEO protest event in the sentence at all. To supplement the CAMEO-based approach in these settings, we generated a list of the most common phrases found in sentences in our training data that contained an anchor for a Violent event. We pruned these by hand (and expanded to related terms) and used them as a supplemental phrase list of violence indicators that also trigger the assignment of a Violence attribute.

3.2.4.4 Location

The assignment of the Location attribute operates as follows:

- For each event mention *without* a Location argument found by the sentence-level event models, we look for a facility, location, or geo-political entity mentioned within a certain distance of the anchors of the event mention. For each anchor, we first look for a location-type mention in the sentence of the anchor, then in the sentences preceding the anchor, and finally in the sentences following the anchor. Only location-type mentions within five sentences of the anchor are considered (this threshold was set empirically on the development data).
- For each event cluster, we select all Location arguments attached to its constituent event mentions. Note that because this process is performed after the document-level splitting and linking stage, an event cluster will not contain multiple places that contradict each other; if it had, it would have been split into multiple events during clustering.
- Finally, we resolve those Location arguments to a gazetteer and produce the most specific (city, state, country) tuple possible.

Resolution to the gazetteer is handled by BBN SERIF. This component was not changed for this effort. It relies on document context to resolve ambiguous places to a gazetteer derived from geonames. If it cannot resolve a place more specifically, it will attempt to at least determine the appropriate country for each location by examining document context.

3.2.4.5 Date

As for Location, we rely primarily on the output of the sentence-level models to identify the appropriate Date arguments for event mentions. We found empirically that this approach led to an over-generation of Date arguments, so we also apply several filters that discard Date arguments that do not meet at least one of several criteria—primarily, the Date argument must typically be syntactically connected to at least one anchor in the cluster, or in some cases the parent or child predicate-argument propositions for an anchor.

Date normalization is performed by BBN SERIF; this component was not changed for this effort.

3.2.4.6 Magnitude (Crowd Size)

Like Population, the Magnitude attribute (which we also call *crowd size*) is primarily derived from the Entity arguments extracted by the sentence-level models. The process is also similar:

- Examine all entity mentions extracted as event participants by the sentence-level models. If any is either modified by or is itself a representation of crowd size, select that. (This component relies on the syntactic parse structure to extract the actual number from phrases like *tens of thousands of demonstrators*, *400 people*, or *a crowd of 10,000*.)
- If nothing is selected above, examine all entity mentions that are co-referent across the document with the entity mentions above. Follow the same guidelines to select an explicit crowd size, if possible.
- If nothing is selected above, look at other entity mentions in the sentences where this event was found. Excluding certain types of entity mentions as ineligible (e.g. law enforcement), select explicit crowd size from entity mentions which are (1) syntactically connected to anchors, (2) likely event participants based on their description (e.g. *activists*), or (3) described as being multiple thousands of people (this is almost always a description of crowd size).

3.2.5 Tuning Precision and Recall

In some contexts, a high-precision or high-recall system might be preferable. We use two parameters to tune our system for precision and recall:

Sentence-level event confidence. BBN KBP produces a confidence score between 0 and 1 for each sentence-level event mention. Our system can use these confidence measures to prune either event mentions (before clustering) or entire event clusters (after clustering). When pruning event mentions, any event mention whose confidence falls below a certain threshold is dropped, before clustering takes place. When pruning event clusters, the system looks at all the sentence-level events that make up an event cluster and prunes only those clusters where no sentence-level

constituent event mention has a confidence that meets the threshold. Our experiments showed that the latter approach was more successful. This is intuitively reasonable: low-confidence event mentions are more likely to actually be correct when surrounded by other, higher-confidence event mentions, with which they may be clustered. In those cases, it is important to keep the low-confidence event mentions in play, since they may contribute meaningful information during attribute assignment. However, if no event mention in an entire cluster has high confidence, it is more likely that this entire cluster is errorful.

Maximum event mention distance. As discussed above, one of the parameters in the clustering algorithm is the maximum number of sentences across which two event mentions may be joined into the same cluster. Setting this parameter higher results in fewer event clusters (lower recall), since more event mentions are allowed to be linked together. Setting it lower results in more event clusters (higher recall).

For this evaluation, we selected three configurations for testing, one tuned for precision, one for recall, and one for F-measure (balanced). Parameters were selected by grid search over the development set. We selected the setting with the top F-measure as our F-Measure setting. We then constrained our search to those settings with .015 of the top F-Measure and selected the settings with the highest precision and recall.

- Precision: confidence-threshold=.3, maximum-link-distance=7
- Recall: confidence-threshold=0, maximum-link-distance=2
- F-Measure: confidence-threshold=.2, maximum-link-distance=3

The top F-Measure setting produced results fairly similar to the one optimized for recall on the development set. This gave us reason to believe that perhaps the selection of that setting might be slightly overtuned to the development set; this proved to be true upon examination of the test set. The effect of these parameters is presented in Section 4.2.2.1.

3.3 BBN ACCENT

The following were the primary tasks required to support the release of BBN ACCENT to the research community:

3.3.1 Extended Input and Output Functionality

The new delivery has two supported modes of operation:

- *Batch mode.* The system starts up, processes a set of files provided to it, and then shuts down. This mode is typically used when processing a large corpus (by running many batches in parallel on a cluster of machines).
- *Server mode.* A python-based client sends document text to a BBN ACCENT server using a HTTP protocol. Typically used for on-demand processing

We provide a sample client for simple use of BBN ACCENT in server mode (sending a single file or a list of files) and direct access to the Python API, if desired.

We provide two output formats: a sqlite database and a new data format called CAMEO XML. Figure 3 provides a sample excerpt of CAMEO XML.

```
<Event id="1" sentence_id="4" tense="neutral" type="1831">
  <Participant actor_id="32187" actor_name="Belgium" agent_id="600"
    agent_name="Armed Gang" role="Source"/>
  <Participant actor_id="32187" actor_name="Belgium" role="Target"/>
</Event>
```

Figure 3: Sample of CAMEO XML

3.3.2 Tools for Data Exploration

BBN ACCENT produces events in the CAMEO ontology, linked to the W-ICEWS actor database. Raw event output is not human-readable, e.g. (192, 38992, NULL, 28813, 173). Under this effort, we added actor and agent names to the CAMEO XML, but the result still a bit opaque. We therefore also developed a tool that displays results in a human-readable format, linked to the original document text. Figure 4 provides a screenshot of this tool. Clicking on the blue text in either window will link to the parallel blue text in the other window.

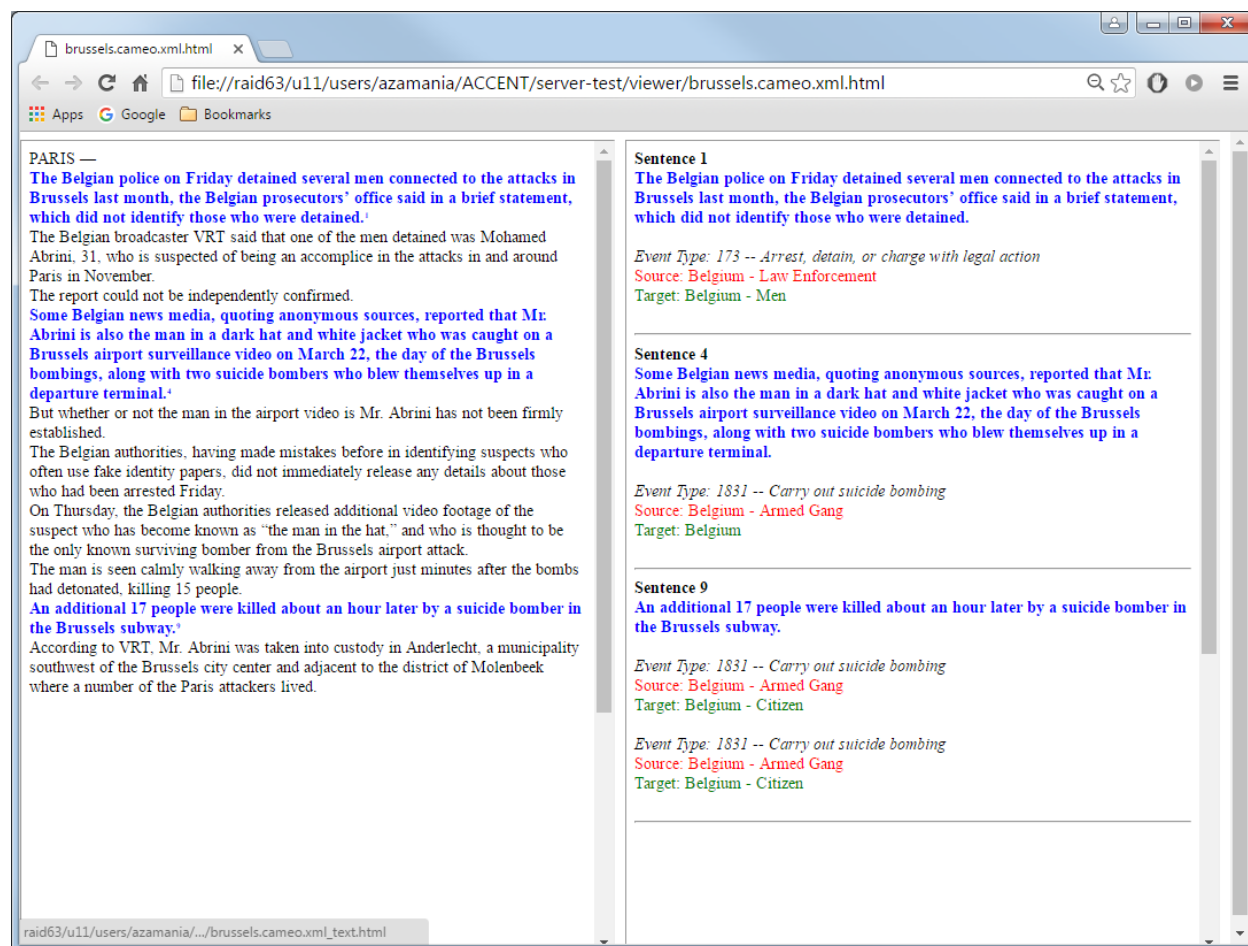


Figure 4: BBN ACCENT data exploration tool

3.3.3 Robust Error Handling & Recovery

Errors in BBN ACCENT are not necessarily reported in a way that is interpretable to non-developers, e.g. “*No region found for sentence; using id=0 for passage*”. It is not at all clear what this means and whether an end user should be concerned. Under this effort, we modified BBN ACCENT’s error reporting to be understandable to a general user, provide better contextual information (e.g. which sentence is involved?), and to suggest possible remedies. A new example of an error message is the following:

The system could not identify a publication date for this document. A publication date is required to resolve relative date references (e.g. 'yesterday' or 'last January'). If this is not important to you (or you have no publication date), you can disable this warning by including 'no_document_date' in the list of warnings specified by the log_ignore parameter (please see Advanced Use). If you would like to specify a document date, this is typically done via XML or SGML metadata (please see Publication Date).

3.3.4 Other Robustness Improvements

We also made other improvements to system robustness under this effort, including:

- Refactoring a core document-reading component to improve handling and calculation of string offsets in various formats
- Fixing one severe memory leak (as well as several minor ones)
- Adding encryption/obfuscation to protect models derived from proprietary BBN data and data released by other parties (e.g. the LDC) under a license that prohibits redistribution

3.3.5 Non-Standard Inputs

BBN ACCENT is designed to operate on well-formed English prose text. In the “wild”, we anticipate it being run on a broader set of documents which might include other languages and non-prose. We therefore modified the system to identify and suppress processing on what appears to be non-English text and added a “prose zoner” stage to remove spans of text that do not appear to be prose text (even if English).

We also were concerned with “in the wild” performance on very long documents. BBN ACCENT processing is non-linear in document length, so long documents are a possible problem. Pre-existing mechanisms at BBN handle this for other tasks by splitting and then re-merging documents during processing, so we extended these mechanisms to include CAMEO events and actors. We also experimented with a different algorithm for entity co-reference (implemented outside of this effort), but this didn’t seem to provide sufficient speed improvements to be worth pursuing in this context.

3.3.6 Performance Customization

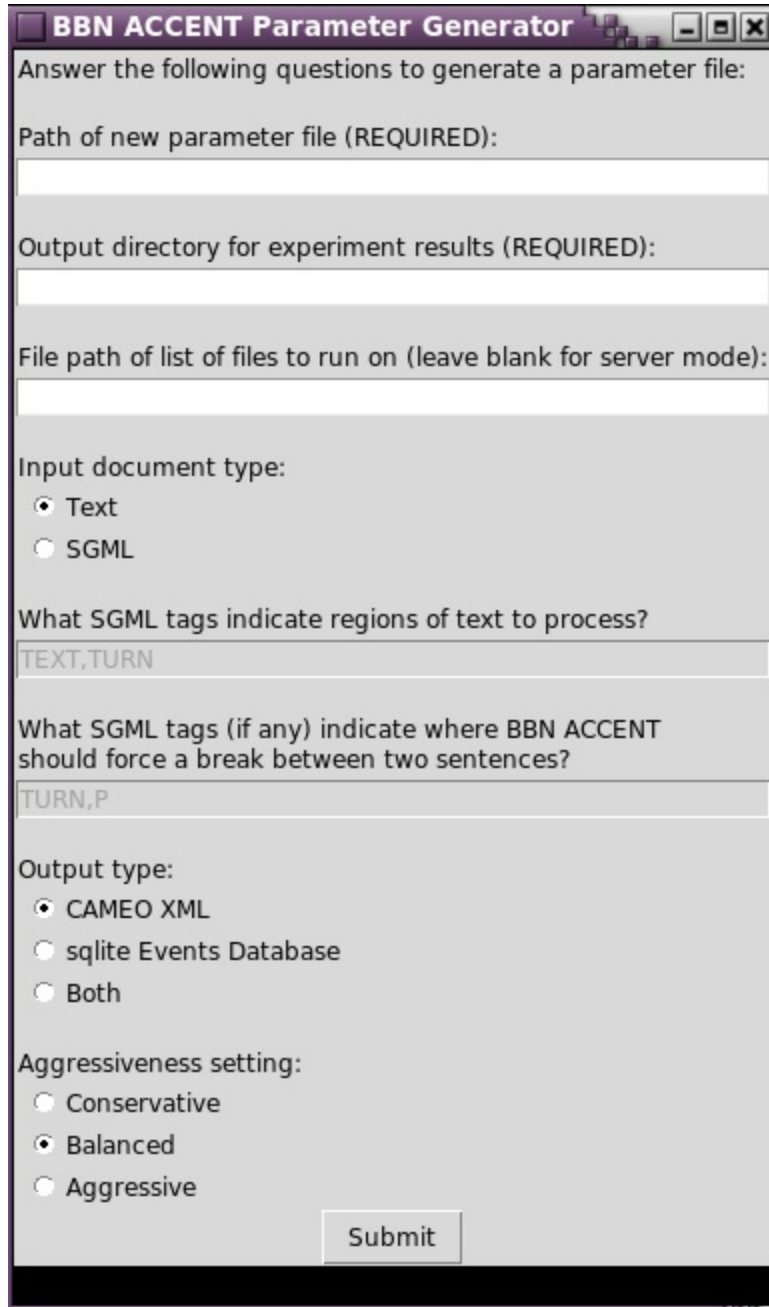
We added a performance customization parameter to tune the system for desired precision/recall use case: high precision (conservative), high coverage (aggressive), or balanced. This parameter

relies mostly on the confidences assigned to actors during the actor coding process to determine which events to report.

3.3.7 Ingest Customization

BBN ACCENT needs document dates (publication dates) to make an informed decision on the tense of each event. To allow this, we added and documented the ability for a user to specify a publication date in server mode. We also tested and documented the means of supplying a document date in other contexts

Further, BBN ACCENT has 300+ run-time parameters; many of these should never be changed by a user and only a few are of interest to an external user. We therefore exposed a small parameter set for user customization, along with detailed descriptions and usage suggestions. To enable easier customization, we developed a GUI for customizing an ACCENT run based on simple input from the user. Figure 5 shows a screenshot of this tool.

A screenshot of a Windows-style application window titled "BBN ACCENT Parameter Generator". The window contains several text input fields and radio button groups. The first three fields are for "Path of new parameter file (REQUIRED)", "Output directory for experiment results (REQUIRED)", and "File path of list of files to run on (leave blank for server mode)". The "Input document type" section has two radio buttons: "Text" (selected) and "SGML". The "What SGML tags indicate regions of text to process?" field contains the text "TEXT,TURN". The "What SGML tags (if any) indicate where BBN ACCENT should force a break between two sentences?" field contains the text "TURN,P". The "Output type" section has three radio buttons: "CAMEO XML" (selected), "sqlite Events Database", and "Both". The "Aggressiveness setting" section has three radio buttons: "Conservative", "Balanced" (selected), and "Aggressive". A "Submit" button is located at the bottom right of the form area.

BBN ACCENT Parameter Generator

Answer the following questions to generate a parameter file:

Path of new parameter file (REQUIRED):

Output directory for experiment results (REQUIRED):

File path of list of files to run on (leave blank for server mode):

Input document type:

☒ Text

☐ SGML

What SGML tags indicate regions of text to process?

TEXT,TURN

What SGML tags (if any) indicate where BBN ACCENT should force a break between two sentences?

TURN,P

Output type:

☒ CAMEO XML

☐ sqlite Events Database

☐ Both

Aggressiveness setting:

☐ Conservative

☒ Balanced

☐ Aggressive

Submit

Figure 5: ACCENT Parameter Generator

3.3.8 Installation Package

We wrote installation instructions and comprehensive documentation for the software package.

4 Results and Discussion

4.1 Novel Event Class Discovery

4.1.1 Automated Evaluation

4.1.1.1 Similarity Metric

Within each cross-validation fold, we train the same/different classifier (S/D classifier) and the similarity measure optimizer on the training data, tune their parameters on the development data, and perform predictions on the three test datasets.

To evaluate the same/different classifier and the similarity measure optimizer directly, we take the 3000 instances in the Accented data set and produce a same/different prediction for each of the ~4.5 million (3000 choose 2) instance pairs. The S/D classifier produces this prediction organically; for the similarity measure optimizer (which returns a similarity real value between [0.0 – 1.0]), we simply treat values ≥ 0.5 as a *same* prediction, and a *different* prediction otherwise. We can then directly calculate Recall, Precision, and F1 scores over those pairs, using ACCENT’s output as a gold standard:

$$\text{Recall} = \frac{\# \text{ true-positive pairs}}{\# \text{ positive pairs in dataset}}$$
$$\text{Precision} = \frac{\# \text{ true-positive pairs}}{\# \text{ pairs predicted as positive}}$$

Table 6 shows these results:

Table 6: Pairwise prediction results. Results are averaged across the 4 folds. Note that the F1 presented above is the average F1 across the 4 folds, and not calculated from the averaged recall and precision.

	Recall	Precision	F1
S/D Classifier (intermediate evaluation)	0.286	0.244	0.255
Similarity Optimizer (intermediate evaluation)	0.335	0.408	0.364
Similarity Optimizer (final evaluation)	0.423	0.510	0.455

Our experiments in the intermediate evaluation consistently showed that the similarity metric optimizer achieved better automated scoring results than the S/D classifier in all evaluation settings (including the downstream clustering evaluation). The similarity optimizer is a natural fit for our task of predicting similarity between pairs of event instances: we simply provide features based off of each event instance and calculate a similarity score. On the other hand, the S/D approach is based on learning a classifier that uses only features computed over pairs of events (as it classifies a pair as same/different event class) and then using that classifier’s probability estimate of a pair of events belonging to the same event class. We found it challenging to create predictions using an S/D classifier that were useful for classification. Standard classifiers functions minimize loss derived from accuracy on the training data. However, with a class balance of 10%/90% (positive/negative), models that attain high accuracy and high F1 may have very different parameters. To maximize the performance of the S/D classifier, we weighted positive training

instances much more heavily than negative ones and selected hyperparameters to maximize F1. However, with the similarity measure optimizer approach, we are able to carefully control the optimization process itself, allowing us to direct optimization such that it achieves the maximum F1. We also found that we were able to use a much larger feature set in the similarity measure optimizer than the S/D classifier without causing overfitting on the training data, enabling us to use many more features. Based on these results, we focused further efforts on the similarity metric optimizer.

4.1.1.2 Full Task

We use the output of the similarity measure optimizer to drive the application of CBC and automatically measure cluster quality using three automated metrics. All three metrics are numbers between 0 and 1, with higher scores representing better performance.

- Normalized mutual information (NMI; Manning et al. 2008) is an information-theoretic interpretation of clustering. NMI is derived by calculating the mutual information between the predicted and ground truth clustering, and then normalizing by their respective entropy.
- BCubed (Han et al. 2012) evaluates the precision and recall for every object in a predicted clustering according to ground truth. These per-object scores are then averaged for an overall precision and recall, from which we calculate the associated F1-score.
- Pairwise F-measure (Manning et al. 2008) calculates an F1-score based on the number of true-positives (TP), false-negatives, and false-positives by comparing the predicted clustering against the ground truth clustering.

While we manually evaluate clusters from both datasets (details in the next section), we perform automated scoring here only on the Accented datasets; the Wild dataset contains too few ACCENT-coded instances (4% out of 3,000 is just ~120 instances) for automated scoring. Table 5 shows these results:

Table 7: Automated scoring of the clusters. Results are averaged across the 4 folds.

	Pairwise-F	BCubed	NMI
Intermediate evaluation	0.334	0.315	0.516
Final evaluation	0.429	0.379	0.539

4.1.2 Ablation Experiments

In our work, we tuned our hyper-parameters and performed feature selection on Gigaword 2002 (our development set). We then performed feature ablation experiments on Gigaword 2003 (our test data). Since we must rely on automatic scoring, we test only on *Accented* data, omitting individual feature types while training on the remaining features. We then recompute the pairwise prediction results from the similarity optimizer and the clustering metric scores. For instance, in the row “WordNet features” in the tables below, we train on all features except the WordNet features.

Table 8: Feature ablation results on pairwise predictions. For the reader’s convenience, we also show the results from using all features

Features		Pairwise similarity scores		
Feature category	Feature type	Recall	Precision	F1
Use ALL features		0.423	0.510	0.455
Predicates in TG	ALL minus Word embeddings	0.307	0.327	0.314
	ALL minus WordNet features	0.420	0.512	0.454
Proposition roles in TG	ALL minus Direct roles & roles on path	0.435	0.504	0.459
	ALL minus Role sequence patterns	0.438	0.496	0.457
	ALL minus TG structure substitution	0.420	0.512	0.454
Argument	ALL minus Sector overlap	0.488	0.467	0.471
	ALL minus Neighboring entity types	0.449	0.494	0.462

Table 9: Feature ablation results on clustering. For the reader’s convenience, we also show the results from using all features.

Features		Clustering scores		
Feature category	Feature type	Pairwise-F	BCubed	NMI
Use ALL features		0.429	0.379	0.539
Predicates in TG	ALL minus Word embeddings	0.228	0.233	0.363
	ALL minus WordNet features	0.436	0.381	0.538
Proposition roles in TG	ALL minus Direct roles & roles on path	0.437	0.382	0.538
	ALL minus Role sequence patterns	0.426	0.379	0.538
	ALL minus TG structure substitution	0.417	0.375	0.536
Argument	ALL minus Sector overlap	0.407	0.370	0.526
	ALL minus Neighboring entity types	0.440	0.386	0.537

First, we observe that the *word embeddings* feature provides the highest utility among all feature types, as omitting it gives the largest reduction in performance (for instance an F1 drop of 0.455 to 0.314 in Table 8). From the clustering results, other useful features include *TG structure substitution* and *sector overlap*. Interesting, while *sector overlap* provides a slight improvement in clustering, it is detrimental to the pairwise predictions. One reason is the recall/precision tradeoff. We theorize that including *sector overlap* gives a boost towards precision (but at a cost of recall) which is actually beneficial towards clustering. The other feature types are either negligible or slightly hurts performance.

It is interesting to note that the three features with highest utility (word embeddings, TG structure substitution, and sector overlap) each come from a different feature category (of which there are also three). This likely implies that dropping any entire feature category will result in a loss of performance, but we did not directly test this.

In Table 10 below, we separately analyze the effects of using document vectors. We show how our pairwise and clustering scores changes, as we train using all features but give different weights to the document level context in Equation 2 (page 13). In our primary system, we set $1 - \alpha = 0.05$, thus conservatively giving only a slight (overall 5%) weight to the document vectors. If we

were slightly more aggressive ($1 - \alpha = 0.1$), we obtain slight improvements to the clustering results.

Table 10: Results when giving different weights to the document vectors.

$1 - \alpha$ Weight	Pairwise scores			Clustering scores		
	Recall	Precision	F1	Pairwise-F	BCubed	NMI
0.0	0.430	0.504	0.456	0.421	0.373	0.540
0.05	0.423	0.510	0.455	0.429	0.379	0.539
0.1	0.415	0.515	0.452	0.438	0.387	0.548

4.1.3 Manual Evaluation

To further assess the performance of our approach, and to generate diagnostic information about the CAMEO ontology itself, we also gathered manual judgments on the clusters produced by the similarity measure optimizer in the constrained condition.

4.1.3.1 Annotation Tool & Process

We adapted an existing BBN annotation tool to allow a user to load a set of candidate event instances, drag and drop them to form a set of human-generated clusters, and label the resulting clusters. Figure 6 shows a screenshot of partially completed annotation for a set of candidate event instances. (In real use, the tool is typically maximized to show more buckets at a time than are shown in this example.)

In this example, twenty candidate event instances were provided to the annotator. These initially appeared in the panel on the left. (**Bold type** indicates the Source actor and underlining indicates the Target actor.) The annotator has dragged and dropped several of the instances to the “buckets” on the right, adding and deleting buckets as desired and creating labels for each bucket (e.g. *visit*, *speak-negatively-about*, etc.). There is also a MISCELLANEOUS bucket on the far right, in which the annotator has placed instances that (in the annotator’s judgment) are not sufficiently “eventlike” (that is, they do not belong in a bucket at all).

To evaluate the quality of a system-generated cluster, we first randomly sample twenty instances from that cluster. We then ask an annotator to separate these instances into “buckets” using the annotation tool. If the system performed perfectly, the annotators would always place all instances from a system-generated cluster into a single bucket—this would mean that the system-generated cluster was entirely cohesive. However, since we expect our system-generated clusters to be significantly more noisy than this, the annotation tool is an excellent way to force annotators to think through (and label) what types of events really are represented in the cluster.

We do *not* ask annotators to maintain a consistent set of bucket labels throughout the entire annotation process, i.e. across multiple system-generated clusters. To do so would enable us to automatically calculate full clustering metrics (e.g. separation as well as cohesion), but doing so in a way that does not introduce bias would effectively require annotators to view all instances for all clusters for a given experiment at once, dealing with many hundreds of instances at a time. This would be very difficult to do effectively, let alone efficiently. Instead, to evaluate separation, we take approach described in the following section.

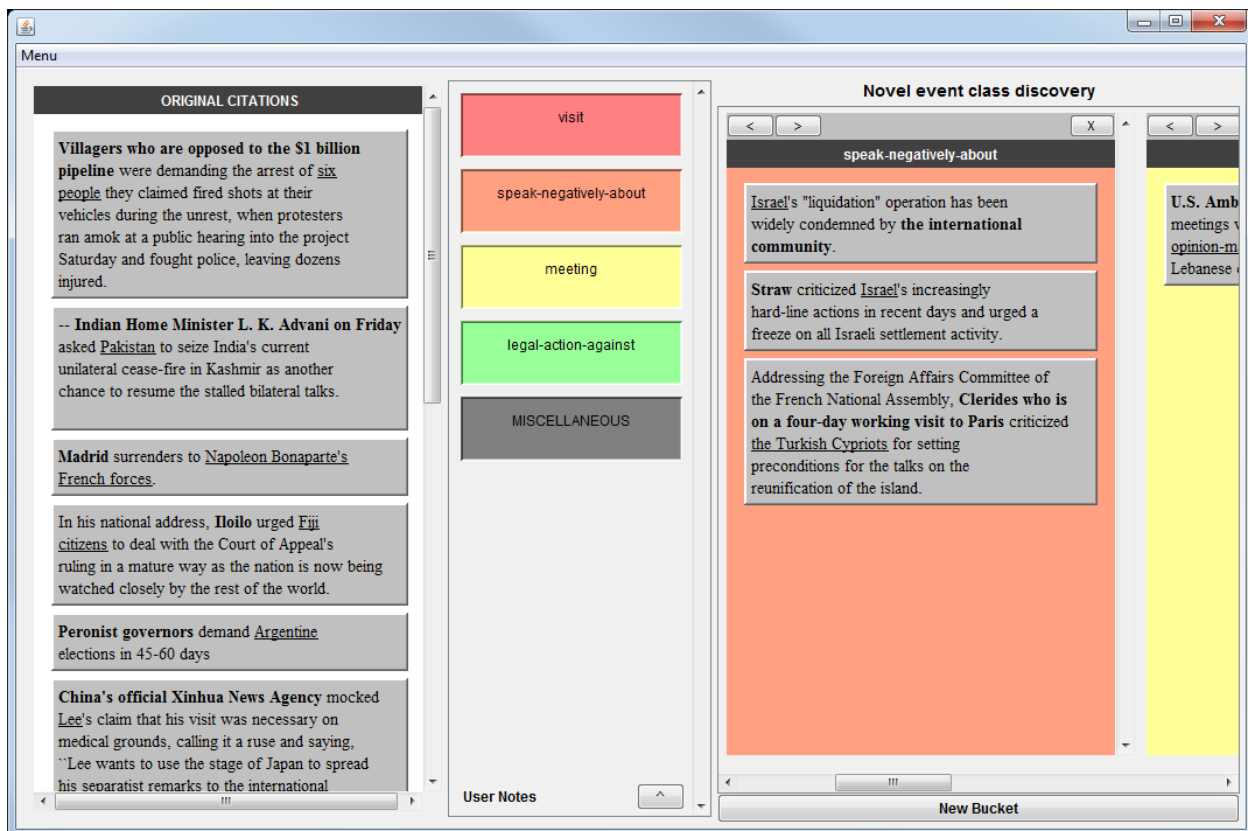


Figure 6: Sample Screenshot of BBN clustering tool

4.1.3.2 Evaluation Metrics

4.1.3.2.1 Cohesion

Given a system-generated cluster that has been judged by a human, we can produce the following metrics:

- **Size of dominant class (as judged by the human annotator).** For each cluster, what percentage of its instances belong to the largest bucket created for it by the annotator? For instance, out of the twenty proposed instances, the annotator might create three buckets—visit (12 instances), meet (4 instances), and criticize (1 instance)—and place three instances in Miscellaneous. In this case, the dominant class would be visit, and the percentage of total instances it represents is $12/20 = 60\%$.
- **Size of dominant class (as judged by BBN ACCENT).** What percentage of instances would belong to the largest bucket, if we grouped instances according to the code assigned by BBN ACCENT? This metric is really only meaningful in the condition where all event instances have ACCENT codes and so is only presented in that case.
- **Cohesion on a scale of 1-5.** As described in the proposal, we measure cohesion on a scale of 1-5. This is derived directly from the size of the dominant class as judged by the human. A cluster whose dominant class represents fewer than 20% of its instances receives a score of 1, a cluster whose dominant class represents fewer than 40% receives a score of

2, etc. We present both the average score as well as the number of clusters receiving a score of 3 or higher (the threshold described in the proposal).

- **Size of miscellaneous class.** How many of the instances were deemed not-eventlike by the annotator?

4.1.3.2.2 Separation

Our approach to evaluating separation is as follows:

- Randomly sample pairs of clusters judged to be at least moderately cohesive (3+ on the scale of 1-5). (Clusters must come from the same cross-validation fold.)
- For a given pair of clusters A and B, merge and randomize their instances. We call this cluster AB.
- Ask an annotator to sort the instances of AB into buckets, just as they would in the standard cohesion task.
- For each bucket that the annotator creates, identify the number of instances that belong to A and the number that belong to B. For each bucket, consider the magnitude of its overlap to be the smaller of those two numbers. So, for a new bucket with 14 instances from A and 2 instances from B, its overlap is 2.
- Sum the overlap from all buckets; call this V .
- We consider the final overlap score for clusters A and B to then be $\max\left(\frac{V}{\text{size}(A)}, \frac{V}{\text{size}(B)}\right)$.

In the worst case where A and B actually both represent the same code, all instances will be put in a single bucket. Here, $V = \min(\text{size}(A), \text{size}(B))$, so the overlap score will be 1. In the best case where A and B are entirely separable, $V = 0$, so the overlap will be 0. In a more complex case, take the following simple example:

- Cluster A: Meet, Meet, Meet, Negotiate, Negotiate, Sue
- Cluster B: Negotiate, SignDeal, SignDeal, SignDeal

Here, the only bucket in common is Negotiate, containing 2 instances from cluster A and 1 instance from cluster B, so $V = 1$. The overlap score is therefore $\max\left(\frac{1}{6}, \frac{1}{4}\right) = 0.25$.

We consider pairs of clusters with overlap scores below 0.2 to be in category 5 (entirely separated), between 0.2 and 0.4 to be in category 4 (mostly separated), etc.

In the Accented condition, we also evaluate clusters using the original codes assigned by Accent.

4.1.3.3 Evaluation Baseline

Before evaluating system output using this method, we would like to understand our baseline. To explore this, we generated ten completely random sets of twenty candidate event instances, all thought by BBN ACCENT to be CAMEO events, i.e. paralleling the Accented condition. We then asked each of our two annotators to process each set into buckets. Table 11 shows the scores received by these randomly-generated clusters.

Table 11: Scores for randomly-generated clusters (averaged across both annotators)

	Human Annotator			ACCENT	
	Size of Dominant Class	Cohesion Score (1-5)	Cohesion Score (#clusters \geq 3)	Size of Miscellaneous Class	Size of Dominant Class
Annotator #1	18%	1.2	0	8%	15%
Annotator #2	18%	1.3	0	3%	15%
AVERAGE	18%	1.25	0	6%	15%

18% is thus a reasonable baseline for size of the dominant class in the Accented condition.

We expected that in the wild the scores would be even lower, as it is even less likely that two randomly selected wild instances will be in the same event class. We performed this same operation generating ten completely random sets of *fifty* candidate event instances (to give the human a higher chance of finding buckets of size greater than two instances). We generated instances only from those that had already been judged to be eventlike, so there were no miscellaneous events (again, simply to maximize the chance that the annotator had to find meaningful groupings). Table 12 shows the results of this experiment for the Wild condition.

Table 12: Scores for randomly-generated clusters in the Wild condition

	Size of Dominant Class	Cohesion Score (1-5)	Cohesion Score (#clusters \geq 3)
Annotator	9%	1.0	0

9% is therefore a reasonable baseline for the Wild condition. (It is actually a generous baseline, given that we removed all non-event-like instances first.)

4.1.3.4 Evaluating System-Generated Clusters

For the final evaluation, we collected the top twenty clusters from each fold for each condition, normalizing the system-generated cohesion estimates by the size of each cluster. The motivation behind ordering the clusters for evaluation in each case is that we expect our final system would *not* suggest all clusters for inclusion in an ontology expansion but rather would automatically identify those most likely to be useful instead and present only those.

As discussed above, we changed our final approach to consider the committees generated by CBC rather than the clusters that span the whole space. We evaluated both approaches for the Wild condition (Wild-AllClusters and Wild-Committees); it is clear the committees provide better (more cohesive) representations of event classes. We therefore only evaluated separation for the second, more cohesive approach. For the Accented condition, we evaluated only the AllClusters approach, simply due to time constraints. (We expect that the Committees approach would also outperform AllClusters for the Accented condition, but it was not evaluated.)

Table 13 shows the overall scores for the experiments. For cohesion score, we present both the average score as well as the number of clusters receiving a score of 3 or higher (the threshold described in the proposal).

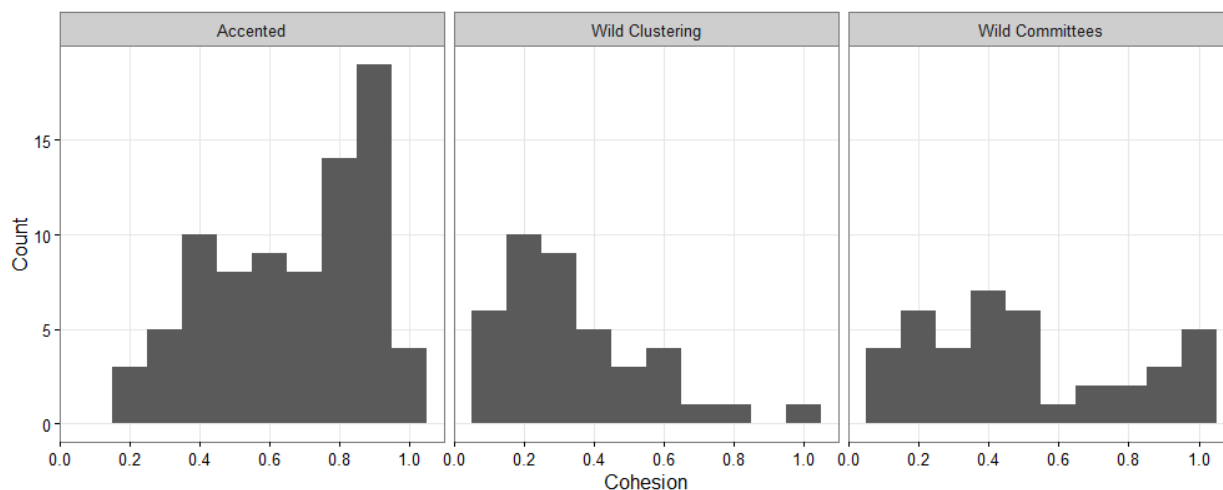
Table 13: Final evaluation scores (cohesion)

	Human Annotator			ACCENT	
	Size of Dominant Class	Cohesion Score (average)	Cohesion Score (#clusters ≥ 3)	Size of Non-Event Class	Size of Dominant Class
Accented	69%	3.8	69 / 80	4%	66%
Wild-AllClusters	32%	2.0	19 / 80	10%	--
Wild-Committees	40%	2.4	31 / 80	10%	--

Table 14: Final evaluation scores (separation)

	Human Annotator		ACCENT	
	Separation Score (average)	Separation Score (#pairs ≥ 3)	Separation Score (average)	Separation Score (#pairs ≥ 3)
Accented	4.7	38 / 40	4.4	37 / 40
Wild-Committees	4.8	38 / 40	--	--

We can also plot the size of the dominant class in a histogram for the three conditions:

**Figure 7: Cluster cohesion (size of dominant class)**

Overall, performance on the Accented condition is obviously much higher than on the Wild condition. As we mentioned above, the wider pool is more difficult for several reasons, including the fact that the event/non-event classifier weeds out some but not all of the noise (as we can see, 10% of candidates are still non-events, even when the E/NE classifier is tuned for precision; the E/NE classifier was not run for the Accented condition). Also, and more significantly, the long-tailed distribution of event classes provides greater challenges in a wider pool. In the limited pool, there were only ~150 classes represented, so a number of event classes had a good number of instances. However, in the wider pool, there seem to be significantly more distinct classes—the pool is dominated by a handful of very common classes and lots of near-singletons. This was the basis for the change from full clustering to committees; as we see, cohesion is significantly higher

after this change. Figure 8 shows this distribution difference demonstrated in the distribution of the similarity metric over the Accented and Wild candidate pools.

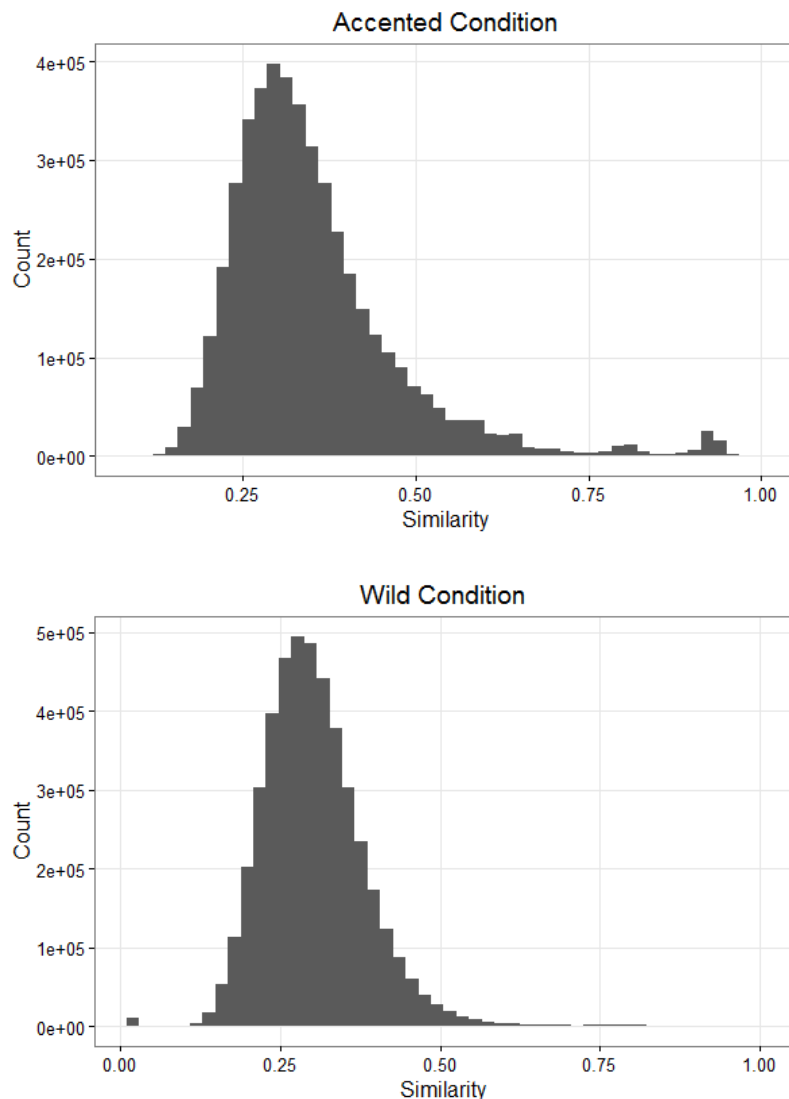


Figure 8: Distribution of pairwise similarity metric output across Accented and Wild candidate pools

In the Accented condition, we see many more high-similarity events, supporting higher-precision clustering. In the Wild condition, fewer instances appear close to each other, suggesting that there are simply fewer sets of highly-similar instances to be grouped together.

Separately, we note that the human annotators' cohesion judgments about the size of the dominant class, on average, appear to line up with the ACCENT judgments (69% vs. 66%). This is generally encouraging, meaning that both are likely to be reasonable metrics by which we can judge performance. We discuss this more in Section 4.1.3.6.

In terms of comparison against proposed milestones, our proposal stated that for the final evaluation, we would attempt to meet two milestones. First, 40% of novel event clusters should be

judged as a 3 or higher in terms of both cohesion and separation (compared to other cohesive clusters). Second, at least ten of the novel event classes discovered by the system should correspond to an event class represented in CAMEO.

The 40% milestones are met clearly in the Accented condition on both counts (86% for cohesion and 95% for separation). They are also met for separation in the Wild condition (95%), and very nearly met for cohesion (39%).

The second milestone is also easily met in the Accented condition, since 86% of all clusters are judged relatively cohesive and correspond to CAMEO categories by the design of the evaluation condition. In the Wild condition, there are so few Accented instances (~100 per fold) that there are not ten reasonably-sized CAMEO clusters available to be rediscovered in each fold. That said, as the folds vary, more than ten distinct CAMEO clusters are certainly rediscovered. For instance, of the 20 instances sampled from the top-ranked cluster in the second fold, 17 shared the same CAMEO code (Request) as judged by ACCENT. Other classes identified across the folds that seem to align to CAMEO include Accuse, BeWarned, MeetWith, Negotiate, Order, TravelTo, MakeDeal, AttackedBy, Instruct, etc.

Examples

Figure 9 shows an example of a system-generated cluster (from the Accented condition) from the intermediate evaluation judged by a human to be >50% cohesive.

In this example, the annotator judged the system to have clustered together two types of activities, one concerning attending funerals and memorials (7 out of 12 instances) and one concerning attending inaugurations (5 out of 12 instances). (This cluster only had 12 instances, so all were evaluated.) In the case of the funerals/memorials, the diversity of expression is encouraging, including *arranging a burial*, *declaring mourning days*, *attending a funeral/memorial*, and *burying someone*. (It is also interesting to note that in this case, CAMEO actually places these two types of events in the same category (017 / Engage in Symbolic Act). So, on CAMEO's terms, this cluster was judged 100% cohesive.)

U.S. Secretary of State Madeleine Albright said after attending the funeral of Assad last Tuesday that she "sensed encouraging signs" in Bashar's remarks on his father's policy toward the regional peace process.

Clarke also said the helicopter entourage that included the war's commander, Gen. Tommy Franks, may have been fired on Saturday as **he** traveled in Afghanistan to attend the inauguration of the new government of Hamid Karzai.

Chaya prayed for help arranging a proper and religious burial for her mother, a very observant, conservative Jew.

Wang, **who** attended President Chen Shui-bian's inauguration in Taipei on May 20, said that "China will have to follow the example of Taiwan in the long run," and that "Taiwan's experience (in implementing democracy) has made me very optimistic about the democratic future of China."

All liquor shops, slaughter houses and cinemas in the country were closed on Friday and Saturday, which **the government** had declared mourning days for the late prime minister.

Bacon said **Cohen** plans to attend a Navy memorial for victims of the Cole attack, likely to be held Wednesday in Norfolk, Va. Bacon said President Clinton also may attend.

Ugandan President Yoweri Museveni left here for Mozambique's capital Maputo Friday to attend the inaugural ceremony of President Joaquim Alberto Chissano scheduled to take place Saturday afternoon.

"We're, however, taking them seriously until it is proved otherwise," said the official in a statement issued in the central town of Dodoma, where **she** is attending the inauguration of the new parliament

Elsewhere, **thousands** attended the funeral of a Palestinian gunman.

Recently, **he** went to New York to attend the funeral of Father Mychal F. Judge, a Franciscan priest killed at the World Trade Center.

Gani Gjaka buried his wife, Nerimane, on Sunday, laying her shrouded body in a muddy field on the southern edge of this tense and bitterly divided town, where Mrs. Gjaka was one of the eight Albanian victims of an eruption of violence by local Serbs.

She arrived in Buenos Aires Wednesday after attending the inauguration ceremony of Peru's new President Alejandro Toledo as a Chinese government envoy, and paying a visit to Colombia.

Recently, **he** went to New York to attend the funeral of Father Mychal F. Judge, a Franciscan priest killed at the World Trade Center.

Figure 9: Example of system-generated cluster (Accented condition)

As mentioned above, we do not ask annotators to maintain consistent labels across clusters. However, one can get a very general sense of whether the clusters are well-separated by looking at the labels assigned to the dominant class. For instance, for the clusters evaluated for the first fold of the Accented condition, Figure 10 shows the dominant labels.

Detain	MakeEmpatheticGesture	Support
PositiveOutlook	Cooperate	Support
Investigate	Cooperate.Diplomatically	Defy
MakeDealWith	Caution	MakeVisit
OppressedBy	DeployTroops	Greet
Sentence	MakeEmpatheticGesture	Investigate
EstablishDiplomaticTies	Indict	

Figure 10: Sample dominant cluster labels

We see some overlap (two instances of Support, two of MakeEmpatheticGesture, two of Investigate) but also significant diversity. Even the overlap is not necessarily a judgment that these clusters are exactly the same, since label consistency was not enforced. For instance, one of the clusters whose dominant label is MakeEmpatheticGesture was shown above in Figure 9. Figure 11 shows a sample of the instances from the MakeEmpatheticGesture bucket in the other cluster where this was the dominant label.

A grim-faced President Bush mourned the deaths of thousands of Americans in Tuesday's atrocities and vowed to avenge their killings.

Noting that the terrorist attacks against the U.S. were unprecedented and shocking, the **PSP leader** expressed her party's deep condolences to the families of the victims.

Murray, though **he** expresses sympathy for goalies.

Japanese Prime Minister Junichiro Koizumi called the attacks ``unforgivable" and expressed his sympathies for the American people.

Sudan, which is listed by the U.S. State Department as a sponsor of terrorism, said in a government statement Monday that **it** rejected terrorism and had extended its condolences to the victims of the Sept. 11 attacks.

In a telegram of condolence, Zhu said **he** was shocked to learn about the crash of the Russian-made MI-8 helicopter and expressed his deep sympathy to the relatives of the victims.

Red-and-white flags flew at half-staff and a carpet of cut flowers outside the royal palace in downtown Copenhagen grew larger Wednesday as **Denmark** mourned the death of its beloved Queen Mother Ingrid.

``I reflect on history and sincerely hope that we will never repeat the tragedy of war," **the emperor** said, offering his condolences to all those who died in the war.

Figure 11: Sample of instances from second MakeEmpatheticGesture bucket

Here, we see that although there is some overlap, this second bucket appears to focus more on the verbal expression of condolences, compared to a more physical act of burial or attendance at a memorial service.

4.1.3.5 Inter-Annotator Agreement

To further understand this evaluation process, we would like to know how stable it is. Specifically, how much will two humans typically agree on which instances should be clustered together?

We attempt to answer this question by making use of thirty-two system-generated clusters which were annotated in parallel by two people for the intermediate evaluation (sixteen in the Accented condition and sixteen in the Wild condition), as well as the randomly-generated clusters generated for the baseline. (However, because many of the buckets in the randomly-generated instance sets contain only a single event instance, this condition may be less meaningful for this comparison—one bucket might look double the size of another, which might show up as a significant change in the size of the dominant class—but probably one bucket just has one instance and the other has two.)

First, Table 15 directly compares the two annotators' results on the primary evaluation metric presented above (size of dominant class), as well as on the percentage of instances they placed in the Miscellaneous class.

Table 15: Comparison of two annotators' results on evaluation metrics

		Size of Dominant Class		Size of Misc. Class	
		Ann #1	Ann #2	Ann #1	Ann #2
System-generated	<i>Accented</i>	74%	72%	2%	3%
	<i>Wild</i>	22%	13%	32%	67%
Random	<i>Accented</i>	18%	18%	3%	8%

As we can see, the two annotators agree most of the time on the size of the dominant class in the cluster. This is encouraging for its suitability as an evaluation metric. However, there was significant disagreement about what goes into the Miscellaneous bucket. It makes sense that this did not affect the size of the dominant class—the instances placed by only one annotator in the Miscellaneous bucket were probably unlikely to match well with whatever the dominant class might be. However, we suspected it would be a significant problem when building an event/non-event classifier, so we iterated several times on guidelines for the event/non-event classification task before performing production annotation.

We can also more thoroughly compare one annotator's buckets to a second annotator's buckets, using our standard clustering metrics, and, given that they tend to agree on average on the size of the dominant class, we can also look further at the Pearson correlation across all clusters in the size of the dominant class. Table 16 reports these results.

Table 16: Annotator vs. Annotator agreement

		Correlation in Size of Dominant Class	Cluster Agreement		
			Pairwise-F	BCubed	NMI
System-generated	Accented	0.81	0.82	0.80	0.68
	Wild	0.54	0.36	0.34	0.50
Random	Accented	0.77	0.55	0.51	0.91

Here, we can more directly see the impact of the disagreement over Miscellaneous, as represented in the significantly lower agreement on the Wild condition (where whether to place an event in Miscellaneous is actually the most significant decision, since relatively fewer eventlike instances end up in the same bucket).

4.1.3.6 Human vs. ACCENT

To further understand both our task and this evaluation process, we ask a final question: How does this manual evaluation process relate to the CAMEO ontology used by ACCENT? Specifically, how close will the annotators come, without any guidance, to recreating the CAMEO ontology in the buckets they create?

This experiment cannot be perfectly completed with the data available here, since the annotators in this case are actually at least somewhat familiar with the CAMEO ontology, which might easily influence their decisions. However, it is still worth examining the question. We can use the same metrics as in the previous section to compare the buckets generated by the humans to the buckets that would have been generated using CAMEO codes provided by ACCENT. Table 17 shows these results, along with the results from the human vs. human comparison for the same dataset repeated for reference.

Table 17: Annotator vs. Annotator/ACCENT agreement for system-generated Accented clusters

	Correlation in Size of Dominant Class	Cluster Agreement		
		Pairwise-F	BCubed	NMI
Annotator #1 vs. ACCENT	0.61	0.77	0.75	0.63
Annotator #2 vs. ACCENT	0.64	0.78	0.76	0.61
Annotator #1 vs. Annotator #2	0.81	0.82	0.80	0.68

On the one hand, it seems that agreement among the humans is somewhat higher than their agreement with CAMEO. This may point to there being some distinctions in CAMEO that do not come naturally to someone who is not a social scientist. On the other hand, the correlation/agreement is still quite reasonable, meaning that the humans are very likely capturing a significant portion of the meaning represented by the CAMEO ontology.

4.1.4 Experiments in Next Steps

4.1.4.1 Ontology Alignment

Novel event classes might be refinements of existing classes, related to existing classes, or totally unrelated to existing classes. How should they be integrated into an existing ontology? Ideally, a system could make placement suggestions and a human could adjust them as desired. We ran a small experiment to evaluate whether our similarity metric could inform possible ontology placement suggestions.

To do this, we:

- Selected four event classes discovered by the system: ReachAgreement, Support, Close, and LegalAction.
- Manually decided which CAMEO class each was nearest to:
 - ReachAgreement → Sign Formal Agreement (057)
 - Support → Praise or Endorse (051)
 - Close → Impose Administrative Sanctions (172)
 - LegalAction → Arrest (173)
- For each novel class N, ranked each known CAMEO class C by the average pairwise similarity score between instances in N and instances coded by ACCENT as belonging to C.

Table 18 shows the three categories judged to be closest to each novel class based on the similarity metric. The category we had previously decided was the best fit is highlighted in red. In all cases it was one of the top three categories selected by the metric. (Note that these codes were excluded from training for the metric used here, so this is not just memorization on the part of the metric.)

Table 18: Closest CAMEO classes for clusters discovered by the system

ReachAgreement	Support	Close (i.e. closing organizations)	LegalAction (sentencing, etc.)
Sign formal agreement	Praise or endorse	Reduce material aid	Arrest
Reject plan to settle dispute	Provide economic aid	Reduce relations	Impose admin. sanctions
Cooperate economically	Provide military aid	Impose admin. sanctions	Fight with small arms

Table 19 shows the three categories judged to be furthest from each novel class based on the similarity metric.

Table 19: Furthest CAMEO classes for clusters discovered by the system

ReachAgreement	Support	Close (i.e. closing organizations)	LegalAction (sentencing, etc.)
Rally opposition against	Demonstrate military power	Praise or endorse	Reject proposal to meet
Rally support on behalf of	Sign formal agreement	Reject	Sign formal agreement
Employ aerial weapons	Reject proposal to meet	Reject proposal to meet	Praise or endorse

The furthest categories are also quite reasonable, e.g. *Praise* being the least likely fit for a class about closing / shutting down organizations.

Although this notional experiment involved only four examples, the “best match” classes were chosen before the similarity scores were calculated and in each case the similarity metric was able to identify the same “best match” as one of its top three choices. (That is, these examples were not cherry-picked as cases where the similarity metric performed well; the metric performed well on all four cases tested.) Much more thorough experimentation is necessary, but this notional experiment provides preliminary evidence that the similarity metric has promise for placing discovered classes within or adjacent to an existing ontology.

4.1.4.2 Event Coding

To assess how much effort would be required to integrate novel event classes into an automatic event coding process, we manually updated BBN ACCENT to produce a sample set of novel event classes. We chose four event classes discovered by the system, two where the class appeared to fit directly into a CAMEO category (MeetWith and Accuse) and two where the class did not (CompeteAgainst and ReceiveMoney).

The process of building a model for BBN ACCENT is an iterative process that begins with sample text graphs, which are manually pruned and expanded. This process requires a varying amount of effort depending on the complexity of the event class: a class like “provide aid” (which can involve many types of actions, e.g. sending troops or setting up a rescue tent) will involve more effort than a simpler class like “impose curfew” (which is highly lexicalized). For this process, we built two models for each class, a “basic” model which used only the text graphs seen in the cluster and an “expanded” model which allowed for simple syntactic and semantic variation on those text graphs and for simple creation of intermediate content nodes (e.g. *money* or *sports_game*) that help expand coverage. Human effort (for both the basic and expanded models) also involves identifying which constraints on a text graph can be removed and which should be included (e.g. all instances of this text graph in our examples have a person actor as their Source; is that a necessary constraint, or no?). For each code the total manual effort for curation and expansion was limited to fifteen minutes of a BBN ACCENT developer’s time.

We note in all cases that the novel models make use of existing BBN ACCENT building blocks that are general to all codes. For instance, BBN ACCENT understands that in the expression “*X and Y officials*” there is likely a reciprocal action involving X and Y; it handles this without any

specific instruction to do so. Or, as another example, BBN ACCENT contains a template construction that, given an initial text graph like “X’s <noun> <preposition> Y” produces a set of related text graphs “X <makes-happen> <noun> <preposition> Y”. That is, if “X’s meeting with Y” is an initial text graph, the model developer can trivially tell BBN ACCENT to expand this to phrasings like “X held a meeting with Y” or “X joined a meeting with Y” and many other related variations.

To evaluate performance, we ran both the basic and expanded models over ~300K documents from Gigaword (January 2010 – June 2010). For each novel class, we then randomly sampled and manually evaluated 50 events produced by each of (a) the basic models and (b) *only* the expanded models (i.e. not by the basic models). We sampled and evaluated these separately to ensure that each was sufficiently well-represented in the evaluation process. We also evaluated 50 standard CAMEO Accuse events and 50 standard CAMEO Meet/Negotiate events, for comparison purposes. Note that for this evaluation, we did not consider actor resolution correctness, e.g. whether a pronoun is resolved to the correct real-world actor, since entity co-reference is not a part of the task at hand. (However, if the system presented an “actor” that was no actor at all, e.g. thinking “millions of rubles” was a group of people, this was considered incorrect.)

Table 20 below presents precision as evaluated for this experiment. We also report the estimated number of correct events generated for each model M, based on the raw number of events generated ($NumProduced_M$) and the evaluated precision (P_M) for that set:

$$EstCorr_M = P_M * NumProduced_M$$

(We round the estimated number of correct events to the nearest ten to reflect that it is an estimate.) The reported Basic+Expanded precision is a weighted average. For instance, for ReceiveMoney, the system produced 356 events using the basic models (at 92% precision) and an additional 230 events using the expanded models (at 86% precision). The total precision for that class is therefore:

$$P_{B+E} = \frac{EstCorr_B + EstCorr_E}{NumProduced_B + NumProduced_E} = \frac{.92 * 356 + .86 * 230}{356 + 230} = .896$$

Table 20: Evaluation results for event coding experiment

	Basic		Expanded (only)		Basic + Expanded (or full BBN ACCENT)	
	Prec.	EstCorr	Prec.	EstCorr	Prec.	EstCorr
CompeteAgainst	100%	1420	98%	420	99.5%	1840
ReceiveMoney	92%	330	86%	200	89.6%	530
MeetWith (novel)	92%	620	96%	13770	95.8%	14380
Meet/Negotiate (CAMEO)	--	--	--	--	92.0%	20300
Accuse (novel)	100%	4810	92%	440	99.3%	5260
Accuse (CAMEO)	--	--	--	--	94.0%	5120

As we can see, precision for all classes is high. (These levels are typical of BBN ACCENT output once the confounding factor of actor resolution is removed.) In fact, many of the errors are only tangential to the determination of the event class, particularly actor extraction errors (e.g. finding “the office building” as an actor for a meeting) or event modality errors (e.g. “X denied that he

received money from Y” is not a valid ReceiveMoney event by standard CAMEO guidelines). Other errors do include the occasional misinterpretation of context, e.g. “*Obama charged the EPA with restoring the bay to health*” is not an accusation-type of charge.

We can also see that the additional expansion effort can make a small or large difference, depending on the type of event. The effect is most dramatic for the MeetWith class. Most of this is due to typical manual text graph expansions which expand nominalizations (e.g. “*X had a meeting with Y*”) to related verbal constructions (e.g. “*X met Y*”) and which allow constructions like “*X verbed Y*” to be equally represented as “*X and Y verbed*”. For meetings, these two expansions are obviously crucial. Most of the other new events come due to curated synonym expansion: for instance, the automatically-discovered text graphs for CompeteAgainst provided two predicates “*match*” and “*qualifier*” in a construction like “*X’s noun against Y*”. For the expanded model, we then added *game*, *competition*, *cup*, *tournament*, *meet*, and *friendly*. More significant expansion work would of course move beyond these simple kinds of transformations, but they are powerful in and of themselves.

We can also see that the system achieves a reasonably high recall compared to BBN ACCENT. For MeetWith, the recall is ~70% of what BBN ACCENT finds for codes 040 (Consult) and 046 (Negotiate). Some of this can be explained by differences in definition—for instance, briefing reporters is considered a valid 040 CAMEO event, but that is not something the novel event model targeted (at least not intentionally). However, much of it is no doubt due to variation in the way that events are expressed; the novel models likely do not capture the less common phrasings.

For Accuse, it looks like the system actually produces more events than BBN ACCENT, but on closer inspection, this appears to deal with a difference in definition. Specifically, the instances produced by the novel event system included “*Prosecutors allege that Liu made several illegal investments*”. Based on this, the expanded models included synonyms for *allege* like *charge*, as in “*prosecutors charged the man with treason*”. However, it turns out that in CAMEO, a formal charge is actually coded as a 173 (Arrest) event, so it is not included in the number of events found as 112 (Accuse). We can still see that recall for the novel Accuse models—difference in definition aside—is very good, since even using the basic models (which did not include judicial charges), the system finds almost as many (94%) events as BBN ACCENT. We suspect that Accuse is a highly lexicalized code—many things are expressed relatively simply and unambiguously, e.g. as “*X accused Y*” or “*X alleged that Y*”. One set of instances we did see only in the BBN ACCENT output was more ambiguous constructions like “*X said that Y [did something bad]*”. BBN ACCENT does have a meta-level concept of a “bad action” that it uses to help identify instances like this, but the novel event models did not make use of it.

We cannot of course evaluate recall for ReceiveMoney and CompeteAgainst in this framework, but a brief anecdotal examination of the data suggests that the ReceiveMoney model is already retrieving many of the possible events in this class, but that CompeteAgainst could be expanded much more significantly given greater effort.

The ultimate goal of this experiment is to help demonstrate the level of effort that would be required to integrate a novel event class into an event coding ontology and coding workflow. The results in this section show that with a small amount of (expert) human effort per class (<20 minutes), a novel event class could be aligned with an ontology and reasonable baseline results

could be produced. Further human effort could of course improve performance, particularly for recall. Future work could also explore how to achieve the same level of improvement without requiring any system expertise. The pattern curation process for BBN ACCENT currently requires knowledge of BBN SERIF and the details of its text graph matching process, but it could be fruitful to explore transforming many of the principles underlying the pruning and expansion process into a guided process requiring no developer expertise—for instance, the system could automatically suggest rephrasings of a text graph, along with discovered examples in a large corpus, to which a non-expert could say yes or no. Similarly a process could guide a user through the kind of template-based transformations described above (which currently require developer knowledge), e.g. asking “*current models find events of the form “X did something with Y”; is “X and Y did something” a valid rephrasing?*”.

Future work could also examine how to leverage the full set of clusters (or all other event candidates combined with their pairwise similarity scores with respect to those in the target cluster) to produce a fully automatic process, or to produce higher recall with a similarly low level of human effort. End users might also be served by a small amount of human effort to define or expand the “correct” boundaries of the new event class—for instance, the Accuse cluster now contains instances of formal judicial indictments, but should it? A system might be able to concisely suggest such boundary cases to aid this process. Finally, observation during this process suggests that another interesting experiment would be to cluster *just* the instances found by a single event class model, to (potentially) produce more fine-grained models—could this help pick up the presumably-meaningful difference between illegal bribes and standard salaries, both of which were seen in instances annotated for the ReceiveMoney class?

4.1.4.3 Expanding Recall

The goal of this effort was to discover new classes, but a promising possible side effect is the use of this approach to expand coverage of existing classes. Specifically, as a part of its clustering, the seedling system often groups known ACCENT ECs together with ECs that were uncoded by ACCENT. These uncoded ECs can represent novel phrasings of an event (that ACCENT has previously missed). Here are two examples of uncoded instances in system-discovered clusters that otherwise primarily contain ACCENT-coded instances from a single CAMEO class:

- Praise/Endorse (051)
 - **Two young men in the public gallery** stood up and shouted their support for Madikizela-Mandela.
 - The United States drew support from **only three other members of the 15-member Security Council**.
- Sign Formal Agreement (057)
 - China could ban cigarette ads following **its** signing of a U.N. anti-smoking treaty.
 - **Toronto** also inked reliever Cliff Politte to a one-year, 845,000-dollar deal.
 - **He** signed Curtis Martin away from the Patriots after he joined the Jets.

Note the difference in domain (sports) for second example: this suggests a particular possible utility for domain shift problems, where one might be able to run this system on a new domain and use it to pick up on novel phrasings of existing events specific to a particular domain. Recall is a

significant known problem for state of the art technology (~30%), so this overall approach could provide much-needed improvements in this area.

4.2 Civil Unrest

4.2.1 Evaluation Data & Metrics

The core Civil Unrest evaluation was run on 400 documents annotated from the OSI-random test set. All OSI-random test documents were coded by two independent annotators.

We also annotated two additional supplemental test sets: Gigaword (103 documents) and OSI-positive (141 documents). These documents were each annotated only by a single annotator.

Finally, secondary evaluations were also performed using a Gigaword-Middle-East set (100 documents) and a WMT2014 set (200 documents), also each annotated by a single annotator.

We describe our set of metrics below, using the following example gold and system output:

Gold Standard

Event GS-1	
Population	General Population
Reason	Housing
Violence	No
Location	Salvador, Bahia, Brazil
Date	2010-05-21
Magnitude	hundreds

Event GS-2	
Population	Labor
Reason	Other
Violence	Yes
Location	<none>, <none>, Argentina
Date	2010-05-22
Magnitude	--

BBN ACCENT

Event BA-1	
Population	Business
Reason	Housing
Violence	Yes
Location	<none>, <none>, Brazil
Date	2010-05-21
Magnitude	hundreds

4.2.1.1 Precision/Recall/F

In this example above, the annotator coded two events (GS-1 & GS-2) and BBN ACCENT found one (BA-1). During scoring, GS-1 and BA-1 will be aligned. However, BBN ACCENT did not get this event completely correct and per the evaluation plan is awarded 4.33 points out of the possible 7, leading to the following values for precision, recall, and F-Measure:

$$\begin{array}{lll} \text{Precision} & = 4.33 / 7 & = 0.62 \\ \text{Recall} & = 4.33 / 14 & = 0.31 \\ \text{F-Measure} & & = 0.41 \end{array}$$

4.2.1.2 Attribute Accuracy

We are also interested in correctness for specific attributes. Here, it does not make sense to take into consideration unaligned events: we are interested in cases where the system found the same basic event as the gold standard, but erred on an attribute. In the above case, we are interested in the fact that in the aligned pair GS-1/BA-1, the system got the Population wrong but the Reason correct. In this context, we don't care about the specifics of the attributes for GS-2, since the system simply did not find that event. So, for this (diagnostic) measure, we simply calculate correctness over each attribute for all pairs of aligned events. Given the simple example above, this generates:

Table 21: Sample attribute correctness

Attribute	Accuracy
Population	0.00
Reason	1.00
Violence	0.00
Location	0.33
Date	1.00
Magnitude	1.00

4.2.1.3 Sentence-level/Document-level Scoring

We can also analyze system performance by ignoring the question of linking and/or splitting events, asking instead how often the system agreed with an annotator that there was an event in a particular sentence. This evaluation has the advantage that it is also possible to cleanly combine the two sets of annotations, simply by taking the union of the sentences in which they found anchors.

This evaluation is not perfect: annotators were asked to mark all reasonable “anchors” for each event, but they were not explicitly told to mark at least one in every sentence that mentioned to an event—so sometimes when an event was mentioned in many sentences, a particular sentence mentioning that event might not have had an anchor marked. Still, for the most part this is a reliable measure of whether an annotator thought a sentence contained an event, and it is certainly a reliable measure on the document level.

In the example above, imagine the following anchors were marked by a human or by the system: GS-1 (sentences 1, 3, and 4), GS-2 (sentence 5), BA-1 (sentences 3 and 4). Here, there are two true positives (3, 4), two false negatives (1, 5), and no false positives. So, precision is 1.0, recall is 0.5, and F is 0.67.

We can also generate an analogous number at the document level.

4.2.1.4 Inter-Annotator Agreement

It is important to understand how consistently the event coding task can be done by humans, since this sets a ceiling on system performance when graded against those same humans.

The following table presents overall inter-annotator agreement for the OSI-random data set (scoring annotator A against annotator B; precision and recall would be reversed if assessed in the other direction):

Table 22: Overall inter-annotator agreement

	Precision	Recall	F-Measure	#Events (A)	#Events (B)	#Events (Aligned)
OSI-random	0.76	0.76	0.76	192	191	163

We can also look at the agreement on a sentence level and a document level:

Table 23: Sentence-level inter-annotator agreement

Evaluated Annotator	Reference	Precision	Recall	F-Measure
A	B	0.91	0.80	0.85
A	AUB	1.00	0.82	0.90
B	A	0.80	0.91	0.85
B	AUB	1.00	0.93	0.96

Table 24: Document-level inter-annotator agreement

Evaluated Annotator	Reference	Precision	Recall	F-Measure
A	B	0.93	0.86	0.89
A	AUB	1.00	0.87	0.93
B	A	0.86	0.93	0.89
B	AUB	1.00	0.94	0.97

As we can see here, annotator B is more aggressive in marking events, compared to annotator A. At the sentence level, B finds 91% of A's events, but A finds only 80% of B's. At the document level, B finds an event in 93% of the documents where A does, but A finds an event in only 86% of the events where B does. At the same time, it appears that each has roughly the same number of unaligned events at the document level (28 vs. 29), so some of the sentence-level differences at least may simply be a question of how aggressively annotator B marked anchors in multiple sentences for a single real-world event.

Agreement on specific attributes was as follows:

Table 25: Inter-annotator agreement for Attributes

	Pop.	Reason	Violence	Date	Location	Magn.	AVG
OSI-random	0.95	0.71	0.90	0.80	0.96	0.94	0.88

Some disagreement on attributes is “real” disagreement, meaning that each annotator looked at the same real-world event and disagreed about the attribute. However, some is simply an artifact of higher-level disagreements. For instance, annotator A considered the following paragraph to describe one event but annotator B considered it two events:

"Murderer!" some among a crowd of 2,000 shouted as they hammered at police barricades surrounding the presidential palace. Hundreds more demonstrated outside the presidential residence.

In this case, the question is whether the palace and residence are in the same city (the guideline for splitting events)—it turns out that according to Google, the latter is in a suburb of Buenos Aires rather than the city itself. This disagreement has a trickle-down effect on attributes: Annotator A’s single event had two crowd sizes annotated, while annotator B had only one crowd size per event. Only one of annotator B’s events can align to annotator A’s event, so whichever it is, the crowd size agreement will be less than perfect.

Some notes on agreement for each of the attributes:

- *Population*: Agreement here was above .95. Every disagreement was a case where one annotator marked a population as General Population and the other marked it as something more specific. There was no confusion between other categories.
- *Reason*: Reason was the attribute with the lowest agreement (.72).
 - Most disagreements (31/46) were cases where annotator A marked the Reason as *Other Government Policies*, while Annotator B marked it as *Other*.
 - An additional 8/46 came from Annotator A marking the reason as something more specific (e.g. *Energy and Resources*), while annotator B still marked it as *Other*.
 - The final 7/46 differences were Annotator B marking something as *Other Economic Policies* (1) or *Unspecified* (6), while annotator A marked it as *Other Government Policies*.
 - We did have annotators do a correction pass of the test annotation specifically for Reason (knowing that agreement was low), but apparently there were still some differences in approach despite the additional guidance we provided. The boundaries between the categories, particularly *Other*, *Other Economic Policies*, *Other Government Policies*, and *Unspecified* are simply quite fuzzy.
- *Violence*: Agreement here was above .90. Most differences (13/16) were due to annotator A marking an event as violent while annotator B considered it non-violent. A spot check indicates these are likely oversights on annotator B’s part, e.g. *People participating in large demonstrations over worsening economic and political conditions*

in Venezuela have been subjected to arbitrary detentions, excessive force and judicial intimidation.

- *Magnitude*: Agreement here was above .90. Apart from a handful of oversights, many of these apparent disagreements involve larger disagreements on the number of events or their exact boundaries, as discussed above.
- *Date*: Event dates are often implicit and can therefore be quite challenging to assign consistently.
 - Overall, Annotator B was consistently more aggressive in assigning dates: In 25/37 cases, annotator B marked a date for an event and annotator A did not (for 60% of these, annotator B decided the event could be inferred to take place on the document date). For instance, the two annotators disagreed as to whether September 27 was a valid date for the protest in this paragraph: *A mass grave has been found on the outskirts of the Mexican town of Iguala, where 43 students went missing on 27 September. The group had travelled to the area to take part in a protest over teachers' rights.* Only in two cases was annotator A was more aggressive.
 - Apart from a few higher-level disagreements, an additional 5/37 disagreements were matters of specificity, e.g. one annotator marked September and the other marked September 22.
- *Location*: Agreement on Location was above .95; the few errors were mostly errors of specificity, e.g. one annotator marked Kiev and the other Ukraine.

4.2.2 Core Evaluation Results

4.2.2.1 Overall Results

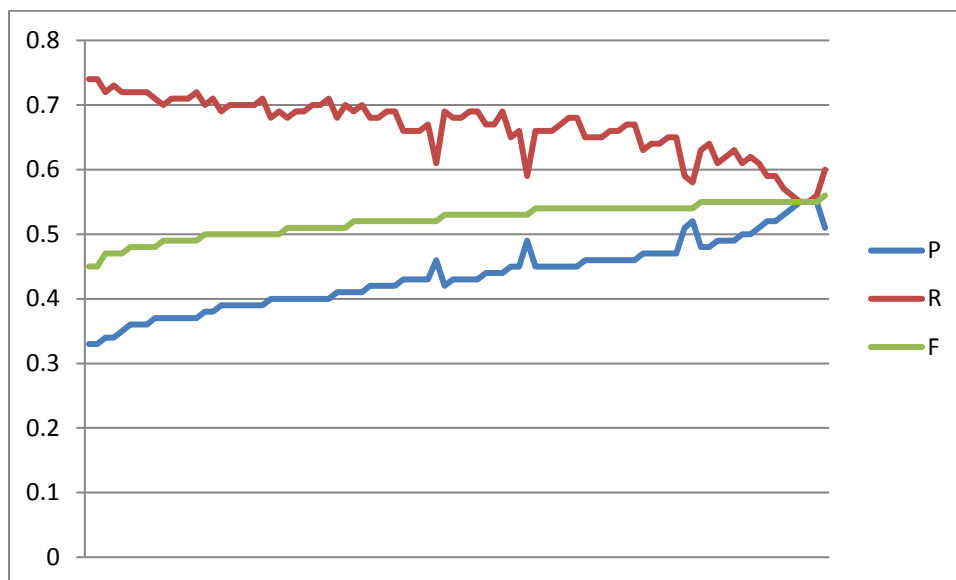
As discussed above, we evaluated three versions of our system, one tuned for precision, one for recall, and one for balance. The following table presents the overall BBN ACCENT results for the OSI-random test set. Since the test set was double-annotated, the system was compared to each annotator's gold standard separately and the results were averaged. For reference, we also include scores for the system as it existed for the January intermediate evaluation.

Table 26: Overall results

Tuning Condition	Precision	Recall	F-Measure	F-Measure (% of Human)	#Events (ACCENT)	#Events (GOLD)
Precision	0.43	0.68	0.53	69%	299	192
Recall	0.33	0.74	0.45	59%	435	192
Balance	0.37	0.71	0.49	64%	370	192
January	0.37	0.54	0.44	57%	278	192

As we suspected, it appears that the *Balance* setting was in fact somewhat overtuned to the development set, and is significantly out-performed by the *Precision* setting. We see here that all settings produce significantly more events than the gold standard; this is true despite the fact that the best-performing setting for the development set produced very close to the correct number of events over that whole set. This kind of tuning could easily be made more robust by simply annotating more development data, a relatively low-cost effort. For this effort, we used the development data both for model development and our final grid search, which is not optimal in a real deployment.

The following graph represents a plot of performance as the tuning parameters vary (sorted by increasing F-measure). For the most part, as F-measure increases, precision increases and recall decreases. The highest recall is .74, the highest precision is .55, and the highest F-Measure is .56.

**Figure 12: Precision, recall, and F-Measure as tuning parameters vary (sorted by F-Measure)**

4.2.2.2 Document & Sentence-Level Results

We can also examine sentence-level and document-level scores. Here, it seems most informative to compare against the union of the two annotators, especially in a context where we might use this system to help a machine create a gold standard—if even one annotator found an event in a sentence or document, it is certainly something we want to find and present for review.

Here, only the first tuning parameter (event mention confidence threshold) is relevant, since the second controls the linking and splitting behavior which is not evaluated by these metrics.⁵ Table 28 gives the results for the different settings of the event mention confidence threshold.

At its most effective (judged by F-Measure), the system is able to recover 90% of the documents in which at least one annotator found an event, with only a 19% false alarm rate. Performance is slightly lower at the sentence-level (recovering 84% of sentences with a 23% false alarm rate). Taken to one extreme—pruning nothing—the system’s document-level recall is 96%. Taken to the other—pruning away any event clusters with a maximum confidence below .9—its document-level precision is 88% (still with a recall of 79%). Of course, these extremes could be increased if that were the goal—these parameters were still fundamentally implemented with the goal of maximizing F-measure—the question is just what the trade-off would be.

Table 27: Sentence-level and document-level Performance

Score Threshold	Sentence			Document		
	P	R	F	P	R	F
0	0.65	0.89	0.75	0.64	0.96	0.77
.1	0.65	0.89	0.75	0.65	0.95	0.77
.2	0.68	0.88	0.77	0.70	0.93	0.80
.3	0.70	0.88	0.78	0.72	0.92	0.81
.4	0.72	0.87	0.79	0.74	0.92	0.82
.5	0.73	0.86	0.79	0.76	0.92	0.83
.6	0.74	0.85	0.79	0.77	0.90	0.83
.7	0.77	0.84	0.80	0.81	0.90	0.85
.8	0.78	0.81	0.79	0.84	0.86	0.85
.9	0.80	0.75	0.78	0.88	0.79	0.83

Here are the same results in graphical form, which highlight the fact that the system’s event mention confidence estimation is very well-behaved: both precision and recall monotonically increase/decrease (respectively) as the value of the threshold increases.

⁵ There is some trickle-down effect of the second to the first, since the first is applied after linking, but it is minimal. The numbers presented here use a moderate value (5) for the maximum linking distance parameter.

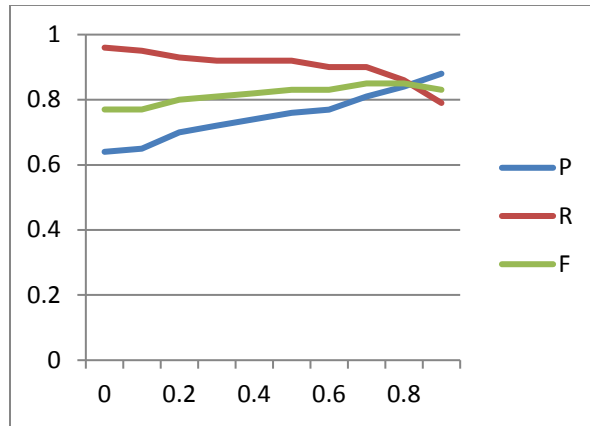


Figure 13: Document-Level precision, recall, and F as event mention confidence threshold increases

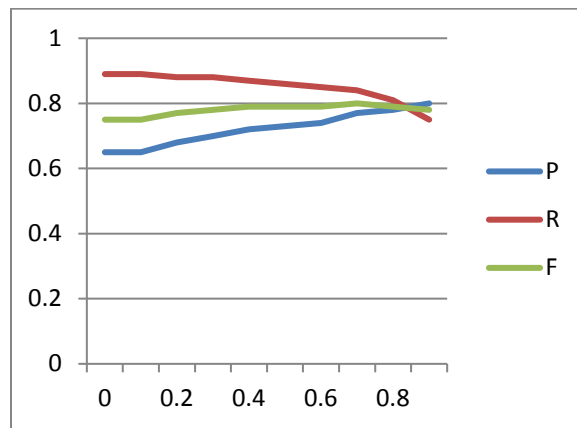


Figure 14: Sentence-Level precision, recall, and F as event mention confidence threshold increases

It is also informative to compare system performance to an individual annotator's. For fairness here, we use the pre-selected "Balanced" set of tuning parameters for the system, though that is not the highest-performing setting:

Table 28: Human and system performance compared against a single annotator

Ref.	System/ Human Evaluated	Sentence				Document			
		P	R	F	% of human	P	R	F	% of human
A	System	0.57	0.90	0.70	82%	0.60	0.92	0.73	82%
A	B	0.80	0.91	0.85		0.86	0.93	0.89	
B	System	0.66	0.91	0.76	89%	0.67	0.95	0.79	88%
B	A	0.91	0.80	0.85		0.93	0.86	0.89	

Here, we see that in terms of recall, the system actually does better than the less-aggressive human. Precision is lower, but we can see by comparing these scores with the scores against the annotators' union above that many of the so-called false alarms are actually cases where the system found a sentence or document marked only by the other annotator. Even with that arguably artificially low

precision, percentage of human performance is on average around 85% at both the sentence and document level.

4.2.2.3 Attribute Results

Table 30 presents the system's agreement with the gold standard (on aligned events) for on each attribute type:

Table 29: System accuracy for attributes

Tuning Condition	Pop.	Reason	Violence	Date	Location	Magn.	AVG
Precision	0.89	0.70	0.81	0.54	0.70	0.83	0.75
Recall	0.89	0.71	0.82	0.54	0.72	0.82	0.75
Balance	0.89	0.70	0.82	0.54	0.71	0.82	0.75

Again, we can also report these numbers as a percentage of human performance:

Table 30: System accuracy for attributes, as a percentage of human performance

Tuning Condition	Pop.	Reason	Violence	Date	Location	Magn.	AVG
Precision	94%	98%	90%	68%	73%	88%	85%
Recall	94%	100%	91%	68%	75%	87%	85%
Balance	94%	98%	91%	68%	74%	87%	85%

Because the tuning parameters only affect which events are included or linked together, their impact on the attribute correctness is minimal.

In terms of percentage of human performance, Population, Reason, Violence, and Magnitude are all very high (close to or well above 90%). The two bigger challenges are Date and Location, which we discuss below.

4.2.2.3.1 Location

There are two challenges in Location detection. First, detecting the correct location in the text, and second, resolving it to the correct gazetteer entry.

Comparing the system to annotator A (for simplicity), about 40% of the errors come when the system identifies the wrong location for an event. For instance, the system thought that the correct location for the following event was Russia instead of the Ukraine:

Yanukovych triggered massive street protests against his government last November when he shunned an anticipated economic pact with the EU in favor of a \$15 billion bailout by Russia.

The remaining 60% of the errors are errors of specificity. In some cases, the system identifies the country only, instead of the city. For instance, in the following sentence, the system located the event in Madagascar, but did not use the information from the third sentence to locate it more specifically in Morondava:

China said it is shocked by violent protests last week at a Chinese-owned sugar plant in Madagascar. The Chinese Embassy in Antananarivo blamed Madagascar officials for not doing their duty to protect Chinese interests. Last week, at least four people died in clashes between police and workers at the plant in the town of Morondava.

In some cases, the reverse is true and the system is more specific than the annotator. Sometimes this is correct. For instance, in the following paragraph, the annotator marked only Taiwan as the location (only looking at the second sentence), but the system located it more specifically in Taipei (from the first sentence):

Demonstrators in Taipei expressed fear in March and April that China was using its economic might to control Taiwan. At the time, Taiwan's parliament was about to ratify a China-Taiwan trade liberalization agreement, but shelved it as protesters occupied the legislative chambers while tens of thousands gathered outside.

Very few errors involve the correct identification of a place in the document but its resolution to some entirely different place. We did not observe any such cases in the test data. In the development data there was one case where a *San Antonio* in Latin America was erroneously resolved to San Antonio, Texas.

4.2.2.3.2 Date

One problem here is the low inter-annotator agreement. The system's performance against one annotator is .65 but against the other it is .45. Low inter-annotator agreement affects not just scoring but also contributes to noise in the training data which can reduce the effectiveness of the model.

Most of the errors (~75%) are places where the system is too conservative about inferring a date from the local context. Often this is because the date is given a different sentence than the one in which the system found the event, e.g.:

*hong kong students scuffle with police during protests. Hong Kong student activists confronted their territory's chief executive and scuffled with police during the second day of a week-long protest calling for China to allow democratic elections in 2017. Thousands of university students are boycotting classes all week, as part of a campaign of civil disobedience to pressure Beijing. On **Tuesday**, about a dozen students pushed past barriers and rushed toward Chief Executive Leung Chun-ying, who was leaving government headquarters.*

Only in about 10% of cases is the system more aggressive than the annotator (15% of the time with annotator A and 7% of the time with annotator B), though sometimes this was correct:

*The military has made the economy a priority after Thailand fell into recession in **the first quarter** as confidence slipped amid political conflict and anti-government protests.*

In the remaining 15% of cases, the system disagrees with the annotator, occasionally because it is less specific, but usually just due to an extraction error by the system.

4.2.2.4 Clustering + Attribute Evaluation (Diagnostic)

To learn more about our system’s performance, it can be helpful to run diagnostic evaluations that use gold standard information for some portion of the system to isolate performance on other aspects of the system. Here, we ran the system with “gold” sentence-level event mentions. These include anchor phrases, participating entities, locations, and dates. We then ran our system over the results, performing event splitting and linking and attribute assignment (including geo-resolution and date normalization).

Table 31: Comparison of performance using only system output vs. system output seeded with gold sentence-level event mentions and their arguments

Condition	System only			Gold (sentence-level) + System		
	P	R	F	P	R	F
Precision	0.43	0.68	0.53	0.66	0.67	0.66
Recall	0.33	0.74	0.45	0.57	0.72	0.64
Balance	0.37	0.71	0.49	0.60	0.70	0.64

As we would expect, performance goes up when seeding the system with gold sentence-level event mentions, but the overall error reduction is only ~30%, meaning we can attribute a significant part of the total error to the challenges of the splitting/linking task (since we know that attribute assignment accuracy, the other task performed by the system in this setting, is relatively high). It is in this area that we think there is the most possible fruitful future work.

4.2.3 Secondary Evaluation Results

4.2.3.1 Supplemental Data Sets

We also evaluated results on two supplemental data sets: Gigaword and OSI-positive.

Since it seems likely that the tuning parameters selected for success on the OSI-random set would not be identical to those selected for these sets, we selected an additional Gigaword-Balance setting⁶ and an OSI-positive-Balance setting⁷ using the top performing settings on the (quite small) development sets for these domains.

For OSI-positive, all four settings perform about the same in terms of F-Measure (though varying in precision and recall). For Gigaword, the settings tuned to Gigaword indeed perform the best, showing the possible sensitivity of these parameters to the dataset.

⁶ Confidence-threshold = 0.3; maximum-link-distance=10

⁷ Confidence-threshold = 0.2; maximum-link-distance=6

Results for Gigaword (103 documents):

Table 32: Overall results on Gigaword test set

Tuning Condition	Precision	Recall	F-Measure	#Events (ACCENT)	#Events (GOLD)
Precision	0.49	0.57	0.53	203	176
Recall	0.41	0.64	0.50	274	176
Balance	0.45	0.63	0.53	244	176
G-Balance	0.51	0.56	0.54	195	176

Results for OSI-Positive (141 documents):

Table 33: Overall results on OSI-positive test set

Tuning Condition	Precision	Recall	F-Measure	#Events (ACCENT)	#Events (GOLD)
Precision	0.60	0.56	0.58	253	270
Recall	0.51	0.66	0.57	347	270
Balance	0.55	0.62	0.58	303	270
OSIP-Balance	0.58	0.57	0.58	267	270

It is interesting to note that in both these settings the number of events found by BBN ACCENT comes much closer to the correct number of events, as it did in development for the OSI-random set. There may be something quite distinct about the OSI-random test set that we observe such a difference between the estimated and actual number of events.

4.2.3.2 Latin America vs. the Middle East

For this effort, we focused specifically on Latin America. However, this raises the question of how well our models would perform in another area of interest. To estimate this, we collected a set of documents focused on the Middle East. We constructed this document set by selecting documents from Gigaword in the same manner that we selected documents focused on Latin America. Specifically, we selected a subset of Gigaword by using an Indri search query derived from the Lexis Nexis search query in Appendix B of the OSI Handbook, modified by replacing the Latin American country names with names of countries in the Middle East.

The only way in which our system is tuned for Latin America is that its training data is biased toward coverage of events in Latin America. (Though coverage of events outside of this area is certainly also represented.) There are no other resources specific to Latin America used by the system (for instance, the gazetteer used by the system covers the whole world). It is possible that performance could have been slightly improved with a very dedicated, area-specific approach. For instance, we could have identified the most common places reported in this data without geo-resolutions and manually created resolutions for them. (For instance, we noticed during development that Altamira Square, a common gathering point in Caracas, is not part of our gazetteer. We could have added a manual fix for this, but we judged the level of effort this would require was not commensurate with its impact.)

Table 34: Overall Results on Gigaword: Latin America vs. the Middle East (using Gigaword-specific tuning)

Region	Precision	Recall	F-Measure	#Events (ACCENT)	#Events (GOLD)
Latin America	0.51	0.56	0.54	195	176
Middle East	0.32	0.55	0.40	54	31

As we can see, performance on the Middle East data is much worse. We do note that it appears that the makeup of the data is very different: both sets have 100 documents in them, but the Latin America set has 176 events compared to the Middle East's 31. With such a small number of events, it would be possible that the lower performance on the Middle East set is just noise.

However, we performed an additional experiment testing both datasets using an alternative set of models trained using documents annotated prior to this Civil Unrest effort, i.e. data with no bias towards Latin America. This includes data annotated by the community for ACE Conflict.Demonstrate events as well as additional data annotated at BBN for the TAC KBP evaluations. We did not use this data in our evaluation system because it hurt performance slightly on our development set. Using it here (in place of the data focused on Latin America), the results were as follows:

Table 35: Results on Gigaword: Latin America vs. the Middle East, using models *without* training data collected for Latin America

Region	Precision	Recall	F-Measure	#Events (ACCENT)	#Events (GOLD)
Latin America	0.50	0.53	0.52	184	176
Middle East	0.44	0.59	0.51	42	31

Here, we see that the data specific to Latin America helped the results on the Latin America set a small amount (0.52 vs. 0.54). However, it clearly hurt performance on the Middle East data, and when using neutral training data, performance on both sets is comparable.

This experiment, though not necessarily statistically significant, seems to indicate that as long as the training data set is sufficiently neutral, transition to a new area of interest with similar performance is not likely to require significant additional work (area-specific annotation might improve performance to some degree, but is not necessary).

4.2.3.3 Spanish Sources

This effort focuses on English text. However, much data of interest is available originally in languages other than English. Is using machine translation a viable approach to extracting events from this data? How much information is lost if so? Could that information be recovered by developing a native system in the non-English language?

We investigate this question in two parts. First, to investigate the loss involved in using machine translation to process documents before extracting events, we selected a set of documents from NIST's Workshop on Machine Translation. This data set is a parallel corpus across multiple languages, including English and Spanish. We selected all documents that matched one of the keywords used by Virginia Tech to select the OSI-random set and ran the Spanish versions of the

documents through BBN's machine translation system. We then ran our Civil Unrest system on both sets to compare event extraction on fluent English (representing a ceiling for what a perfect Spanish system would do) to noisy machine-translated English (representing the current best performance of our system on Spanish data).

Table 36: Comparison of performance using fluent English vs. machine translation

Condition	Fluent English			Machine Translation		
	P	R	F	P	R	F
Precision	0.45	0.47	0.46	0.50	0.47	0.48
Recall	0.39	0.60	0.47	0.38	0.51	0.43
Balance	0.42	0.55	0.48	0.42	0.49	0.45

Performance is actually remarkably comparable (even better in MT on one condition), meaning that machine translation may be a very reasonable pathway for automatic event extraction of this type. Performance on attributes is also very similar between fluent English and machine translation. (Note that Date is not evaluated here, since these documents do not have publication dates against which to normalize.) It is probably very helpful that OSI events are evaluated on normalized attributes rather than strings of text (e.g. proper names), which might be missing or garbled in the machine translation.

It is of course possible that performance on fluent English could be more easily improved using features that rely on syntax or other things often garbled by MT. It is also true that this data set is quite small, so conclusions should be held lightly. So, beyond this experiment, we also attempted to consider whether a native Spanish language coder could be a reasonable option to supply higher performance. Parts of BBN SERIF are currently available in Spanish, including tokenization, name-finding, and parsing. However, performance is not as high as in English (there has been less investment by the government in Spanish, and there are fewer generic linguistic resources available to the models). In addition, there are also some BBN SERIF components that are not available in Spanish, e.g. time and place resolution. BBN ACCENT is also currently an English-only system and so would not be available for use as features for BBN KBP or for the violence attribute assignment process. We could find ways to supplement or work around some of these absences—for instance, we could apply the English geo-resolution component to Spanish; we have previously done this for person and organization names with only a small drop in performance. Still, there would be work to be done. For languages other than Arabic, Chinese, and Spanish, there would be significantly more work to be done, as fewer or no core BBN SERIF components may exist.

Given this, and given the results above, it seems likely that the most effective path could be a hybrid path using both a native Spanish language coder as well as an English coder run over machine translation. Initial experiments (under separate funding) showed that although building a purely Arabic event coder for CAMEO would be very challenging, even a proof of concept Arabic system (with an overall F-Measure lower than BBN ACCENT over machine translation) was able to recover some events lost in the noise of machine translation. We expect a similar situation would exist for Spanish and civil unrest.

4.2.3.4 Comparison against Manual Annotation Process

One important final question is how an automatic system could best be used in the process of actual gold standard creation. The accuracy level of the automatic system on its own is insufficient (at least for now), but could it still improve the efficiency of the overall process?

To attempt to test this, we had two humans perform parallel tasks. One annotated a document “from scratch”, without any machine promptings. The other annotated a document given the output of the BBN system. The time the process took for each document was recorded.

Typically, we perform annotation in ENote, but that format cannot easily be seeded with system output. So, we set up a process modeled directly on the GSR, where annotators are asked to fill in GSR entries for documents (using Excel). Where system output is available, we pre-fill in those events and provide an HTML file showing visually where in the document the system found a particular event (i.e. its anchors, etc.). We performed this task using the output of the system tuned for balance (which is actually heavy on recall, as we saw above). We had annotators alternate whether they were working on a set with or without system output to avoid learning curve bias as best as possible. Each annotator coded 140 documents for this experiment.

One challenge with this experiment was that we have two annotators trained for this task, but they operate at very different speeds (one is consistently faster than the other). Given the difference in average speed between the two annotators, we cannot compare raw numbers (annotator B was faster on every batch, no matter the condition). However, we can compare a change in the speed ratio between the two annotators when they are given different material to work with. When annotator A was given the raw documents (with no system output), they were 1.96 times as slow as annotator B (who was given the documents with system output). When annotator B was given the documents with the system output and annotator A was given the raw documents, they were only 1.57 times as slow. From this perspective, it appears that providing system output did help speed up the annotation process.

Another factor is that the majority (69%) of these documents have no events (according to the annotators). In most cases, the system does not produce any events for these documents either. But in this experiment, the annotator must still carefully read each of these documents—the system’s opinion that the document contains no events is ignored and irrelevant.

We therefore simulated a second experiment, where we assumed that annotators skipped all documents in which the system found no events. What would be the impact on their overall speed and overall recall? We tested this across our event mention confidence thresholds. With the most conservative pruning threshold, this approach resulted in a 44% reduction in time taken while still maintaining a recall of .94. With the most aggressive pruning threshold, the approach resulted in a 62% reduction in time taken while maintaining a recall of .88:

Table 37: Recall and time saved when skipping documents without system events

	Events Found	Recall	Time Spent (Minutes)	Time Saved
ALL	160	1.00	1320	--
0	151	0.94	733	44%
0.1	151	0.94	733	44%
0.2	151	0.94	695	47%
0.3	150	0.94	674	49%
0.4	149	0.93	653	51%
0.5	146	0.91	611	54%
0.6	145	0.91	605	54%
0.7	143	0.89	562	57%
0.8	141	0.88	534	60%
0.9	140	0.88	501	62%

This of course does not reflect any particular effort to maximize performance for this task, so we expect there would be ways to better optimize the system to support this kind of approach. It also does not account for any gains from the actual event mention finding itself (e.g. where in the document the system believes an event to appear, or what its attributes might be)—this would require integration with a real annotation tool to assess.

4.3 BBN ACCENT

All software improvements were completed on schedule; for details see section 3.3.

5 Conclusions

5.1 Novel Event Class Discovery

The Novel Event Class Discovery seedling provides a proof of concept: novel event class discovery via a trained similarity metric and clustering is possible. Below, we discuss recommendations for future work.

5.2 Civil Unrest

The goal of this work was to build a model that extracts civil unrest events at a level of accuracy sufficient to replace (or significantly minimize) human effort. Our overall system achieved 69% of human performance on the entire task, and performed even more effectively when judged on the sentence or document level. Confidence estimates were shown to be reliable and can be used to tune the system for precision or recall. Experiments also showed that a shift in region of interest is possible without retraining and without performance degradation, and that performance on Spanish machine translation is comparable to performance on fluent English. Initial experiments show that this system could be very effective in reducing the level of effort required to create a gold standard event record.

5.3 BBN ACCENT

BBN ACCENT was delivered to the government in September 2016 and is now available for release to the research community.

6 Recommendations

6.1 Novel Event Class Discovery

Three particular challenges stand out as a result of our analysis. First, one significant challenge facing the system appears to be sorting out which predicates on a predicate path are important. In the case of “lend aid” vs. “lend voice”, the direct object is really the distinguisher. In the case of “voiced concern” vs. “dismissed concern”, it’s the verb. Sometimes it is both, sometimes only one. Sometimes the most relevant predicate isn’t even on the path between the Source and Target. Much previous work has often focused on similarity between just two predicates, but this is harder. Our next steps would need to focus specifically on this challenge, looking at novel ways to combine embeddings for multiple words, ways to embed predicate groups/paths rather than just words, the assignment of importance weights to predicates (possibly leveraging IR techniques), and other approaches.

A second challenge for an eventually operational system is the question of granularity. Granularity is something both systems and humans struggle with when creating or managing ontologies. The BBN system is trained on CAMEO codes, so theoretically it defaults to a CAMEO granularity—but this varies a great deal from code to code. An interesting next step would be to look at truly hierarchical (and/or overlapping) clusters that allow it to capture multiple levels of granularity and complexity. (The committees produced CBC do overlap, but we did not focus on this for this effort.)

Finally, a third important challenge for a full system would be the move from discovery to operational event coding. Most trained event extraction algorithms require negative examples. We could of course assume other clusters are negative examples, but this will be noisy and incomplete. In addition, the clusters of positive examples may also be biased or incomplete: just because the system identifies “SellTo” as a new event type doesn’t mean that cluster contains all (or even most of) the ways of expressing that event class. Techniques worth exploring here might include automatic techniques to supplement positive and negative examples as well as semi-supervised active learning techniques (e.g. that present a human with likely borderline cases for annotation).

In summary, the Novel Event Class Discovery seedling provides a proof of concept: novel event class discovery via a trained similarity metric and clustering is possible. Still, there is much work yet to be done: improving the similarity metric & clustering, more complex clusters, pursuing next steps after discovery (ontology placement and automatic event coding). Explorations of other uses of the technology are also worthy of further effort, such as an exploration of its application to improvement of existing event coding models. We hope to pursue each of these as opportunities allow.

6.2 Civil Unrest

It is clear from our experiments that the BBN Civil Unrest system could be of significant use in reducing the level of effort required to generate an accurate gold standard event record. An excellent next step would be the integration of this system with a real end-to-end GSR-generation pipeline; we are exploring this under separate funding. Another important next step would be to explore widening the set of events targeted by a system: we expect that similar results could be achieved targeting a gold standard record of violent activity, for instance. Further work would be necessary to assess (and to minimize) the effort required to move to each new event type.

6.3 BBN ACCENT

The most important next step here is to publicize the release of this newly-available tool. We will work with IARPA, AFRL, and others to spread the word about its availability and to encourage its use in the research community.

7 References

Marco Baroni, Georgiana Dinu and German Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In ACL-2014.

Jiawei Han, Micheline Kamber, and Jian Pei. Data mining: Concepts and techniques. Third edition. Morgan Kaufmann. 2012.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. Introduction to information retrieval. Cambridge University Press. 2008.

Patrick Pantel and Dekang Lin. Discovering word senses from text. In ACM SIGKDD 2002.

Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In ACL-1994.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013.

Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In ICML vol. 14, pages 1188-1196. 2014.

Rong-En Fan and Kai-Wei Chang and Cho-Jui Hsieh and Xiang-Rui Wang and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. In Journal of Machine Learning Research. Vol 9, pages 1871-1874. 2008.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

BBN ACCENT	a BBN event coding system, based on BBN SERIF
BBN KBP	a BBN event coding system, based on BBN SERIF
BBN SERIF	BBN's core natural language analysis suite
CAMEO	Conflict and Mediation Event Observations
CBC	Clustering by Committee
EC	event candidate
ECP	event candidate pair
ENE	event/non-event
ICEWS	Integrated Crisis Early Warning System program (DARPA)
NLP	natural language processing
OSI	Open Source Indicators program (IARPA)
S/D	same/different (event classifier)
TG	text graph
W-ICEWS	Worldwide Integrated Crisis Early Warning System program (follow-on to ICEWS, funded by OSD through ONR)
WN	WordNet (an English lexical database)